PROGRAMACION EN BASIC (y II)

NUEVAS TECNOLOGIAS

BIBLIOTECA DE ELECTRONICA/INFORMATICA





BIBLIOTECA DE ELECTRONICA/INFORMATICA

PROGRAMACION EN BASIC (y II)



Esta obra es una nueva edición actualizada y ampliada de la obra originalmente publicada por Marcombo, S.A. de Boixareu editores, con el título de «Aplicaciones de la Electrónica»

El contenido de la presente obra ha sido realizado por Marcombo, S.A. de Boixareu editores, bajo la dirección técnica de José Mompin Poblet, director de la revista «Mundo Electrónico»

ISBN 84-7634-485-6 (Obra completa)
ISBN 84-7634-810-X (Vol. 48)
D. L.: B. 35621-1986

Impreso y encuadernado por
printer industria gráfica, sa c.n. II, cuatro caminos, s/n
08620 sant vicenç dels horts barcelona 1986

Printed in Spain

Programación en Basic (y II)

LAS CADENAS DE CARACTERES

Si el computador admitiese solamente la programación de variables numéricas carecería de una gran cantidad de posibilidades y aplicaciones. La necesidad de representar gráficamente en la pantalla las variables numéricas hizo necesario el tratamiento de los denominados originariamente *strings* o cadenas de caracteres.



La programación tiene que ser posible incluso para los niños, ofreciendo facilidades tanto para el computador como a las órdenes del lenguaje.

Con ellas se pudieron tratar las palabras, no sólo en plan de almacenar texto sino también analizar sintácticamente las frases, encadenar distintas palabras para formar otras nuevas, reconocer las veces que están en un texto o su

frecuencia y, en consecuencia, manejar nombres y apellidos de gran ayuda en el campo de la gestión administrativa, confeccionando nóminas de salarios, ordenando puntuaciones obtenidas en una determinada prueba junto con la ordenación alfabética.

Las cadenas de caracteres pueden contener letras o caracteres de texto, números, signos algebraicos, signos de puntuación, signos especiales, etc. siempre comprendidos en el programa entre comillas, por lo que estas comillas no pueden figurar como caracteres de las cadenas.



La programación de computadores se facilita con los mensajes que presenta la pantalla del computador.

Se asignan a variables iguales a las numéricas que tienen el signo de dólar para que se distingan de ellas. Muchas veces las funciones utilizadas en el tratamiento de cadenas de caracteres tienen también el signo dólar. Así, la variable numérica A(8) es distinta de la A$ (8) y por ello ambas tienen que dimensionarse previamente con DIM para organizar la memoria interna del computador.

A los caracteres tanto numéricos como alfabéticos o de otro tipo, va asociada una cifra que tienen que ser interpretada por el computador. Esta asociación constituye un código que normalmente suele ser el ASCII, iniciales de *A*merican *S*tandard *C*ode for Information Interchange, y

cuando se establece una comparación entre cadenas de caracteres, se basa en el valor del código. Por ello las cadenas de caracteres pueden sumarse y compararse entre sí usando los símbolos de las comparaciones con variables numéricas.

```
10 REM (21)Cadenas de caracteres
20 A$="ABCDEFGHIJKLMNOP"
30 PRINT " LEFT$       RIGHT$           MID$"
35 PRINT"--------    --------        ----------"
40 FOR N=1 TO 8
50 L$=LEFT$(A$,N)
60 R$=RIGHT$(A$,N)
70 B$=MID$(A$,N,7)
80 PRINT L$;TAB(12);R$;TAB(28);B$
90 NEXT N


 LEFT$       RIGHT$          MID$
--------    --------       ----------
A           P               ABCDEFG
AB          OP              BCDEFGH
ABC         NOP             CDEFGHI
ABCD        MNOP            DEFGHIJ
ABCDE       LMNOP           EFGHIJK
ABCDEF      KLMNOP          FGHIJKL
ABCDEFG     JKLMNOP         GHIJKLM
ABCDEFGH    IJKLMNOP        HIJKLMN
```

Figura 3. Listado del programa n.º 21.

A medida que el sistema operativo de un computador contiene más facilidades para el tratamiento de las cadenas de caracteres, aumentan las órdenes y funciones que permiten este tratamiento; generalmente no suelen faltar las siguientes:

LEN determina la longitud de la cadena de caracteres en su valor numérico. VAL especifica el valor numérico de los caracteres de la cadena. POS determina la posición del carácter dentro de la cadena. STR$ convierte un valor numérico en cadena de caracteres. KEY$ compara el valor del carácter de la tecla accionada con el que figura en el programa CHR$ determina el valor del código ASCII del caracter considerado.

Los tres siguientes se utilizan mucho para recortar las cadenas. Así, LEFT$ separa la porción izquierda de N

caracteres de la cadena. RIGHT$ separa la porción derecha. MID$ separa la porción de la cadena de caracteres entre dos valores de su colocación.

Puede verse con el programa n.º 21 de la figura 3 el tratamiento de las cadenas de caracteres. En la línea 20 se asigna a la variabla A$ la cadena de caracteres formada por las 16 primeras letras del alfabeto y con ellas se hace una tabla en la que por la LEFT$ se extrae la porción izquierda de esta cadena, mientras que por RIGHT$ se hace lo mismo pero de la derecha, aumentando cada línea un carácter por el bucle repetitivo de la línea 40. Puede observarse que por MID$ se extraen 7 caracteres a partir del Nsimo.

```
10 REM (22)Utilizacion de MID$
20 A$="OOOOOOOOOMOOOOOOOOO"
30 FOR N=1 TO 20
40 PRINT MID$(A$,N,N)
50 NEXT N
60 END

 O
 OO
 OOO
 OOOO
 OOOOO
 OOOOMO
 OOOMOOO
 OOMOOOOO
 OMOOOOOOO
 MOOOOOOOOO
 OOOOOOOOO
 OOOOOOOO
 OOOOOOO
 OOOOOO
 OOOOO
 OOOO
 OOO
 OO
 O
```

Figura 4. Listado del programa n.º 22.

Otra aplicación de MID $ la podemos ver en el programa n.º 22 de la figura 4, en la que se utilizan cada vez los valores de la variable de control del bucle FOR...NEXT de la

línea 30, uno para la posición y el otro para la longitud, valores que van aumentando hasta 20, y si el resultado impreso empieza a decrecer a partir de la mitad es debido a que se terminan los caracteres asignados a la variable A$.

```
10 REM (23) Utilizacion RIGHT$ LEFT$
20 FOR N=1 TO 20 STEP 2
30 A$="...................*"
40 B$=RIGHT$(A$,N)
50 C$=LEFT$(A$,20-N)
60 PRINT B$+C$
70 NEXT N
80 END

*...................
..*.................
....*...............
......*.............
........*...........
..........*.........
............*.......
..............*.....
................*...
..................*.
```

Figura 5. Listado del programa n.º 23.

La combinación de las órdenes RIGHT$ y LEFT$ permite colocar un carácter determinado dentro de una cadena, tal como puede verse en el programa n.º 23 de la figura 5. Por este programa se coloca un asterisco en la posición Nsima de una cadena de 20 puntos, en el ejemplo cada dos puntos por STEP 2. Para ello se toman N caracteres de la parte derecha y se rellenan con (20 – N) caracteres de la parte izquierda.

El resultado final se consigue con la suma de las dos variables a las que se ha asignado cada trozo.

Este recurso posibilita la confección de gráficos, histogramas, tableros, etc. en los que un determinado carácter está posicionado con exactitud. No se puede perder de vista que es fácil sustituir los puntos por espacios en blanco.

Una aplicación de lo anteriormente expuesto es el programa n.º 24 de la figura 6, por el que deseamos

comprobar el resultado de un *generador aleatorio*. Se trata de presentar sobre una línea la posición de un asterisco en función del número aleatorio generado, con lo que se puede comprobar si esta posición presenta *regularidades*.

```
10 REM (24)Pruebas generador aleatorio
20 INPUT "SEMILLA=";N
30 REM Generador aleatorio
40 N=N*9821+.2113271-INT(N*9821+.2113271)
50 B=INT(20*N+1)
60 A$="...................*"
70 PRINT RIGHT$(A$,B);LEFT$(A$,20-B);B
80 GOTO 40
90 END

...............*....16
............*.......13
.......*............8
..*.................3
..................*.19
...........*........12
*...................1
.........*..........10
...................*20
.....*..............6
.................*..18
.................*..18
.*..................2
.................*..18
..............*.....15
.......*............8
..................*.19
*...................1
.....*..............6
....*...............5
..................*.19
......*.............7
.*..................2
............*.......13
..*.................3
...................*20
..*.................3
...................*20
..............*.....15
..*.................3
...*................4
.......*............8
........*...........9
.............*......14
```

Figura 6. Listado del programa n.º 24.

El programa queda explicado con el diagrama de flujo de la figura 7, y tiene un generador aleatorio que se obtiene multiplicando el número generado anteriormente por una constante a la que se suma un número que tiene muchos decimales, separándose para el nuevo valor aleatorio solamente la parte decimal. Puede comprobarse que no

presenta regularidades en los 34 números generados, aunque no se han presentado más para no alargar el espacio de la figura. Nótese que en la parte derecha se ha impreso el número generado entre 1 y 20 por la línea 50 del programa.

Otras órdenes para el tratamiento de las cadenas de caracteres son LEN, que determina su longitud y de la que se verá una aplicación en el programa que ordena los apellidos de una lista, separando unos valores numéricos con una cadena de puntos variable en longitud para que queden los valores numéricos alineados aunque los apellidos tengan longitud variable. Otra es VAL que da el valor numérico de una cadena de caracteres, mientras que STR$ hace lo contrario. Esta última puede tener una utilidad importante dado que el computador siempre que escribe en la pantalla un número, deja un espacio en blanco a la izquierda por si ha de colocar el signo menos, pero hay casos en que se desea

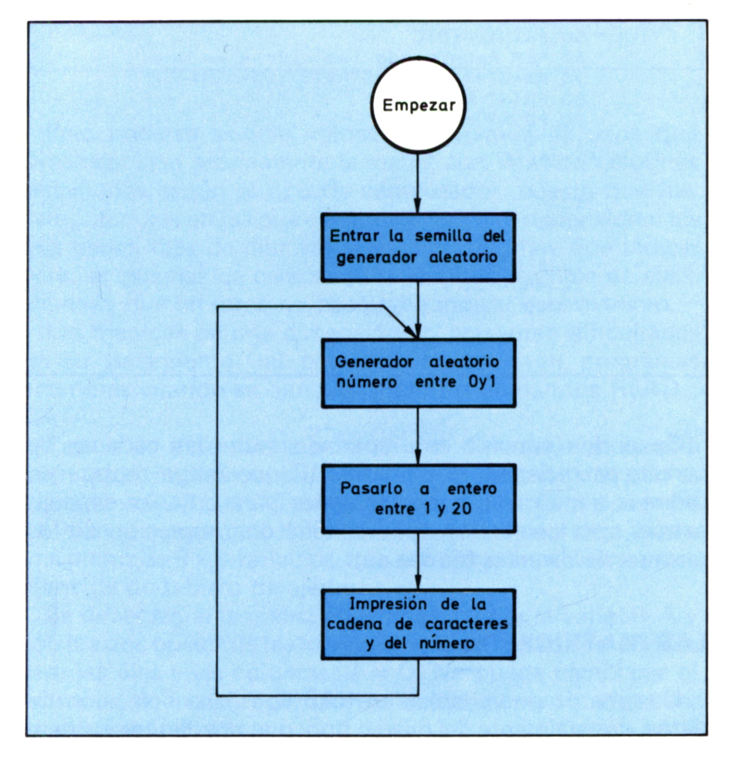


Figura 7. Diagrama de flujo del programa n.º 24.

que no exista este espacio en blanco y la única solución consiste en pasar el número a cadena de caracteres con STR$.

Esto se muestra en el programa n.º 25 de la figura 8, por el que después de entrar tres números en variables numéricas, en la línea 70 pasan a cadenas de caracteres con STR$. Podemos ver tres tipos de impresión por PRINT: el de la línea 50 que pone los tres números con un espacio de separación por si tienen el signo menos delante, el de la línea 80 los presenta sin este espacio y el de la línea 90 realiza la suma de las variables numéricas.

```
10 REM (25)Reunion de tres cifras
20 INPUT"PRIMERA CIFRA=";A
30 INPUT"SEGUNDA CIFRA=";B
40 INPUT"TERCERA CIFRA=";C
50 PRINT A;B;C
60 REM Reunion de las cifras en una
70 A$=STR$(A):B$=STR$(B):C$=STR$(C)
80 PRINT A$+B$+C$
90 PRINT A+B+C
100 GOTO 20
110 END

1234
6543
23
1234 6543 23
1234654323
7800
```

Figura 8. Listado del programa n.º 25.

Se podría concluir este apartado sobre las cadenas de caracteres diciendo que un BASIC que tenga riqueza de órdenes tiene muchas posibilidades para ordenar, separar, extraer, etc. caracteres de la cadena con impresión de los mismos de diversas formas.

LAS MATRICES

Una matriz de variables en BASIC es una colección de datos, generalmente del mismo tipo, que pueden tener una o

dos dimensiones (hay casos de tres) y que puede almacenar valores por órdenes como INPUT o por asignación mediante el signo igual. Las variables pueden ser numéricas o bien de cadenas de caracteres.

Si la matriz es de una dimensión da lugar a una lista, una columna de datos o una fila. Si es de dos dimensiones forma una tabla de valores, determinados por filas y columnas.

```
10 REM (26) Datos tablero 8x8
20 DIM A(8,8)
30 FOR F=1 TO 8
40 FOR C=1 TO 8
50 A(F,C)=0
60 NEXT C:NEXT F
70 END
```

Figura 9. Listado del programa n.º 26.

Para poderse asignar valores a las variables tiene que dimensionarse previamente la matriz con DIM con algunas variaciones según el tipo de computador, puesto que hay computadores en los que solamente hay que dimensionar las que tienen más de diez variables, en otros hay que indicar para las cadenas de caracteres la longitud máxima de ellas, mientras que en otros no hace falta indicar este extremo.

Las matrices de una dimensión no presentan dificultades en su tratamiento, tal como se ha visto en programas anteriores cuando se han explicado los comandos READ... DATA.

Pero si se trata de matrices de dos dimensiones, para la asignación, impresión, etc. tendrán que utilizarse dos bucles repetitivos FOR....NEXT uno dentro del otro. Esto podemos verlo en el programa n.º 26 de la figura 9 el cual deja a cero una matriz de 8 × 8 variables, que por ejemplo podrían ser los datos de un tablero de ajedrez.

Se dimensiona la matriz en la línea 20 por la variable A y con los dos bucles de las líneas 30 y 40 se llenan con el valor cero las filas y las columnas F y C. No puede cambiarse el orden de la línea 60 porque los bucles tienen que quedar uno dentro del otro.

Las matrices de variables tienen muchas posibilidades. Si en el ejemplo anterior en la línea 50 ponemos en vez del valor cero otro valor como el producto de las variables F y C, quedará una tabla de multiplicar en que cada resultado se buscará por los índices F y C, que son el multiplicando y el multiplicador. Poniendo otras expresiones algebraicas relacionadas con las variales F y C se tendrán otras tablas que se imprimirán con facilidad, tal como se ha visto en programas anteriores.



Computador personal de tipo portátil, provisto de un display abatible de cristal líquido. (Cortesía: Philips).

Veremos en posteriores programas aplicaciones de las matrices, sobre todo cuando se han de ordenar sus valores en los programas para ordenar datos que, además de numéricos, pueden ser de cadenas de caracteres, y con ello ordenar listas de nombres, pero siempre hay que comprobar con las instrucciones que acompañan cada computador la forma inicial de dimensionar con DIM, que no es igual para todos ellos.

PROGRAMAS PARA ORDENAR DATOS

Aunque hay computadores que tienen más órdenes de las aquí mencionadas, interesa ver las aplicaciones a la

programación de temas concretos. Empezaremos con los programas denominados SORT, que ordenan los datos de una lista ya sea de variables numéricas o de cadenas de caracteres.

El programa n.º 27 de la figura 11 ordena 10 datos numéricos, si se cambian las variables A y B por A$ y B$ sirve para ordenar cadenas de caracteres, dimensionando solamente la A porque la B es auxiliar.

En la línea 20 está el dimensionamiento por DIM y puede verse que se dimensiona en una variable más, debido a que al ordenar se abandona la primera variable y se ocupa la undécima. Los datos numéricos se entran por el bucle FOR...NEXT de las líneas 40 y 50, que se imprimen sin ordenar por otro bucle de la línea 70 y 80.

```
10 REM (27)Prog.ordenar 10 numeros
20 DIM A(11)
30 REM Entrada de 10 numeros
40 FOR N=1 TO 10:INPUT A(N)
50 NEXT N
60 REM Numeros sin ordenar
70 FOR N=1 TO 10:PRINT A(N);
80 NEXT N:PRINT
90 PRINT "YA ORDENADOS:"
100 REM Subrutina ordenadora
110 FOR N=1 TO 10
120 IF A(N)>A(N+1) THEN 140
130 GOTO 170
140 B=A(N):A(N)=A(N+1)
150 A(N+1)=B:N=N-1
160 GOTO 120
170 NEXT N
200 REM Numeros ya ordenados
210 FOR N=2 TO 11:PRINT A(N);
220 NEXT N
230 END

76 8 234 12 89876 5678 2 12345 456 123
YA ORDENADOS:
2 8 12 76 123 234 456 5678 12345 89876
```

Figura 11. Listado del programa n.º 27.

Después de terminado el proceso de impresión o presentación en la pantalla, el programa va a la subrutina ordenadora de la línea 100, ésta es un bucle FOR... NEXT

que compara cada número buscando el menor, el cual va ascendiendo a las primeras variables hasta que las comparaciones terminan. En las líneas 140 y 150 se halla el intercambio de variables en la búsqueda del valor menor.

Una vez ordenados, los 10 números quedan asignados a las variables de N(2) a N(11) y se imprimen por la línea 210, cuyo bucle repetitivo ya contiene un valor en más.

Respecto a este dimensionamiento es importante citar que hay computadores que permiten hacerlo indirectamente, aplicado a este caso pondríamos DIM A(R+1), siendo R el número de casos a ordenar, disposición muy útil para este tipo de programas pues basta colocar al principio un INPUT R del número de casos a ordenar y ya queda listo para funcionar, siempre que en los bucles FOR...NEXT se haya puesto la variable R en el NEXT.

La impresión de los datos al principio no es necesaria, pero puede servir para comprobar la exactitud del resultado. Con ello se suprimirían las líneas 60, 70 y 80.

Un ejemplo que combina las cadenas de caracteres con la subrutina ordenadora de datos está en el programa n.º 28 de la figura 12. No tiene diagrama de flujo porque está comentado por REM en varias líneas. Se trata de resolver el problema de cualquier oficina que en la selección de varias personas con una puntuación determinada, tiene que hacer dos listas, una ordenada por los puntos que han obtenido en la prueba de selección y otra por orden alfabético de apellidos en la que figuran también los puntos. Este tipo de programa puede tener aplicación a otros casos en que hay que ordenar por datos numéricos y por alfabeto.

Como puede verse en el listado pregunta al principio cuantos nombres va a tener la lista, ya que una vez entrados todos el programa se pone a ordenar inmediatamente, primero por los puntos y luego alfabéticamente. Por ello hay que tener preparada la conexión de la impresora para que no se detenga, aunque podría ponerse un STOP antes de la línea 100 si se desea comprobar las asignaciones a las variables A$.

Tal como está el listado presentará el resultado en la pantalla, pero si se desea imprimirlo hay que cambiar los comandos PRINT por LPRINT u otro que tenga la impresora.

El poder disponer del DIM indirecto hace que por la variable R tengamos solucionadas las rutinas ordenadoras

```
10 REM (28) Ordenar por puntos y apellidos
20 INPUT "CUANTOS NOMBRES=";R
30 DIM A$(R+2):DIM B$(R+2)
40 REM Entrada de puntos y apellidos
50 FOR N=1 TO R:INPUT A$(N):NEXT N
100 REM Ordena por puntos
110 FOR N=1 TO R
120 IF A$(N)>A$(N+1) THEN 140
130 GOTO 170
140 B$=A$(N):A$(N)=A$(N+1)
150 A$(N+1)=B$:N=N-1
160 GOTO 120
170 NEXT N
180 PRINT
200 REM Impresion lista por puntos
210 PRINT "LISTA ORDENADA POR PUNTOS"
220 FOR N=2 TO (R+1):PRINT A$(N):NEXT N
230 REM Pasa puntos despues de apellidos
240 FOR X=2 TO (R+1)
250 L=LEN(A$(X)):B$(X)=LEFT$(A$(X),6)
260 A$(X)=RIGHT$(A$(X),L-6)
270 A$(X)=A$(X)+B$(X)
280 NEXT X
300 REM Ordena por apellidos
310 FOR N=2 TO (R+1)
320 IF A$(N)>A$(N+1) THEN 340
330 GOTO 370
340 B$=A$(N):A$(N)=A$(N+1)
350 A$(N+1)=B$:N=N-1
360 GOTO 320
370 NEXT N
380 PRINT
390 PRINT "LISTA ORDENADA POR APELLIDOS
400 FOR X=3 TO (R+2)
410 L=LEN(A$(X)):B$(X)=RIGHT$(A$(X),6)
420 A$(X)=LEFT$(A$(X),L-6)
430 L=LEN(A$(X))
440 E$="................................"
450 F$=MID$(E$,L-10)
460 PRINT A$(X);F$;B$(X)
470 NEXT X
480 END

LISTA ORDENADA POR PUNTOS
12222 PEREZ VEGA JOSE
12345 RAMOS PEREZ RICARDO
34555 ALONSO RODRIGUEZ JOSE
44444 MARTINEZ MARTINEZ JUSTINIANO
54322 MARTINEZ GOL VICENTE

LISTA ORDENADA POR APELLIDOS
ALONSO RODRIGUEZ JOSE.......................34555
MARTINEZ GOL VICENTE........................54322
MARTINEZ MARTINEZ JUSTINIANO................44444
PEREZ VEGA JOSE.............................12222
RAMOS PEREZ RICARDO.........................12345
```

Figura 12. Listado del programa n.º 28.

que llegan hasta R la primera vez en la línea 110, hasta R+1 al imprimirlo ordenado en la línea 220, al ordenar en la 240 y 310, y finalmente a R+2 cuando se imprime la lista ordenada por apellidos. Si no se dispone del DIM indirecto, hay que poner como líneas de programa las modificaciones de la variable R.



Este computador personal de Atari presenta unas reducidas dimensiones que le permiten fácil acomodo en cualquier lugar de trabajo, además es capaz de entregar impresa la información.

Por la línea 50 se entran los datos poniendo en primer lugar cinco valores de los puntos, un espacio en blanco y luego los apellidos a ordenar, tal como se puede ver en el ejemplo del pie del listado. Como estos datos son una cadena de caracteres, hay computadores que no admiten la coma en el INPUT de cadenas debido a que les indica una nueva cadena a entrar; se tiene que sustituir por un punto, aunque no queda tan bien después de los apellidos y antes del nombre.

Los cinco valores numéricos han de ser siempre los mismos para que queden alineados la segunda vez, pero pueden ponerse ceros a la derecha.

Si son muchos los nombres a ordenar puede ponerse en la línea 50, antes del INPUT, un PRINT N para que en la pantalla vaya apareciendo el número de orden de cada caso y se vea si están todos antes de que empiece la ordenación doble.

Una vez entrados todos los datos se pasa a la subrutina ordenadora por los puntos que en la cadena figuran en

primer lugar, rutina que es igual a la del programa anterior y termina en la línea 170.

El PRINT de la línea 180 está para dejar un espacio en blanco antes de la impresión del resultado de la ordenación y en la pantalla, similar función tiene el de la línea 380.

La impresión de la primera ordenación tiene lugar en la línea 220 y al terminar pasa inmediatamente a la 230, que transporta las cinco cifras de los puntos al final de los apellidos después de haberlos almacenado en B$. Esto es necesario ya que se tienen que volver a ordenar y cada apellido debe llevar sus puntos.

En la línea 300 empieza la nueva ordenación por apellidos, similar a la anterior pero con los valores del FOR...NEXT que aumentan en 1. Esta rutina ordenadora termina en la línea



En los equipos portátiles puede encontrarse idéntica potencia a la que tienen algunos equipos de sobremesa, como sucede con este modelo de Texas Instruments, muy útil para los hombres de negocios.

370, y, pasando a imprimir el encabezamiento de la lista ordenada por apellidos, separa los puntos obtenidos por RIGHT$. Luego los imprime calculando el trozo de línea de puntos con LEN de la línea 430 y se logra con ello que los puntos queden alineados en columna, separados por una línea de puntos de los apellidos, lo cual permite una cómoda lectura de los resultados al buscarlos alfabéticamente.

Si en la matriz de cadenas de caracteres de una dimensión, que supone este caso, es necesario dimensionarla al principio dando el máximo de caracteres con la orden DIM, entonces se supone que constará de unos 35 a 40 caracteres formados por las letras y las cinco cifras. Es necesario tenerlo en cuenta para los computadores que necesitan este dimensionamiento, puesto que de lo contrario se detendrá el programa con el correspondiente mensaje de error.

Dado que el tiempo que emplea en la ordenación puede ser muy largo si el número de individuos es de centenares, la consiguiente espera puede dar la impresión de que el programa no funciona. Por ello, si el computador tiene señales sonoras tipo BEEP, pueden colocarse dentro de la rutina ordenadora y antes del NEXT N, aunque quizás sea mejor intercalar un PRINT N para que se vea en la pantalla como progresa la ordenación.

Las subrutinas ordenadoras son muy útiles para determinados objetivos. Las empleadas aquí no son las más rápidas, aunque son sencillas y fáciles de pasar de un computador a otro. Las mejores utilizan las ventajas de la programación con los nemónicos del código de máquina, pero entonces son para computadores concretos y son necesarias cuando el número de casos a ordenar es elevado.

PROGRAMAS PARA REPRESENTACIONES GRAFICAS

El tema de representar en la pantalla las funciones que procesa el programa está muy relacionado con las órdenes específicas que se añaden a cada computador. La causa de esta falta de unidad tiene que buscarse en el hecho de que no está estandarizado el número de puntos empleados en la pantalla cuando se trabaja en la denominada alta resolución.

Este término se contrapone al de baja resolución, que es el modo normal de representar en la pantalla los textos del programa. Como el número de caracteres que puede

representar la pantalla oscila alrededor de 36 en horizontal y 24 en vertical, las representaciones gráficas en baja resolución pueden servir únicamente para los *histogramas*.

```
10 REM (29) Histograma num.aleatorios
20 DIM A(21)
30 INPUT "SEMILLA=";N
40 INPUT "CUANTOS NUM.=";R:S=-1
50 S=S+1:IF S=R THEN 100
60 GOSUB 200
70 U=INT(20*N)
80 A(U)=A(U)+1:GOTO 50
100 REM Trazado del histograma
110 PRINT"     ==HISTOGRAMA=="
120 B=R/20: PRINT "med.=";B
130 PRINT"     I----------m-----I"
140 FOR C=0 TO 95 STEP 5
150 D=INT(C/5):E=A(D)
160 F=E*10/B:F=15-F:IF F<=0 THEN LET F=1
170 B$="0000000000000000"
180 C$=MID$(B$,F,15)
190 PRINT C;TAB 5;C$:NEXT C:STOP
200 REM Generador aleatorio
210 N=N*2939+PI-INT(N*2939+PI)
220 RETURN
230 END


     ==HISTOGRAMA==
 med.=50
     I----------m-----I
  0  0000000000
  5  0000000000
 10  00000000000
 15  0000000000000
 20  0000000000
 25  0000000000000
 30  000000000
 35  0000000000
 40  00000000000
 45  000000000000
 50  00000000000
 55  00000000000
 60  0000000000000
 65  000000000000
 70  00000000000000
 75  00000000000
 80  00000000000
 85  000000000000
 90  0000000000
 95  0000000000000


210 N=N*7+.13-INT(N*7+.13)
```

Figura 15. Listado del programa n.º 29.

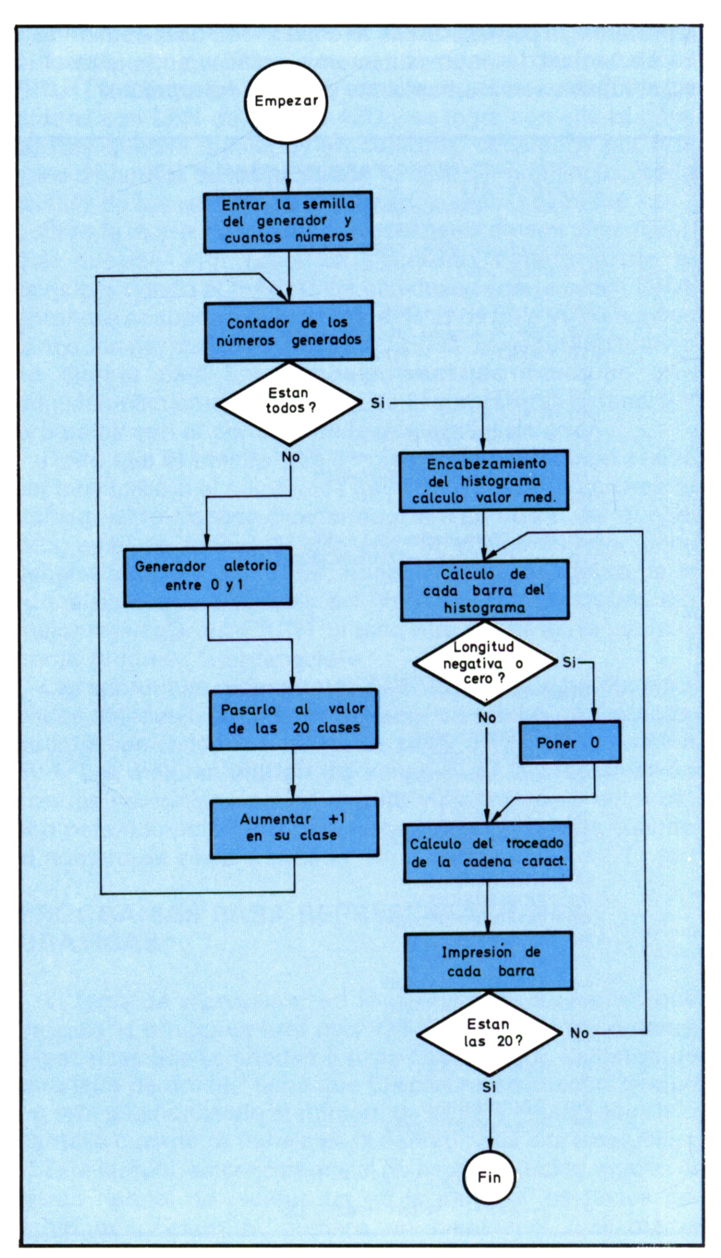


Figura 16. Diagrama de flujo del programa n.º 29.

Por otra parte, la alta resolución permite disponer de centenares de puntos tanto en horizontal como en vertical, puntos que se pueden manejar fácilmente con el programa, entonces las órdenes del BASIC son adecuadas solamente para cada tipo de computador. Es frecuente encontrar el comando PLOT que sitúa un punto a unas coordenadas determinadas, aunque hay que concretar si la pantalla es a color, porque entonces tiene que añadirse el número que tiene el color del punto y el color del fondo, ambos programables. Pero además encontramos toda suerte de órdenes como AT, CURSET, PENUP, MOVE, AXIS, etc., de tal manera que hay que consultar el libro con que se acompaña cada computador para obtener todas las posibilidades que tiene.

Por el manejo de las órdenes normales de BASIC pueden obtenerse representaciones gráficas de histogramas como puede verse en el programa n.º 29 de la figura 15, cuyo diagrama de flujo está en la figura 16. Se trata de generar varios miles de números al azar para comprobar si repartidos en 20 clases quedan distribuidos uniformemente o de lo contrario hay alguna clase con pocos o muchos números.

Para ello, empezamos entrando un número cualquiera denominado *semilla* y también entramos el valor del total de números generados. Se pone en marcha el programa que por un contador comprueba cada vez si se han generado todos los números. El generador aleatorio está en una subrutina al final del programa para poder ser cambiado con facilidad y es del tipo de los ya vistos que entrega números entre cero y uno, no llegando nunca a este valor.

En la línea n.º 70 pasa el número al azar al comprendido entre las 20 clases y por la línea 80 se incrementa en una unidad la memoria de cada una de ellas. Si ya se han generado todos los números se pone el encabezamiento del histograma, se calcula el valor medio de los números de cada clase, se traza el eje con la posición de este valor medio y escribe la longitud de cada barra del histograma según los números que ha tenido.

El valor máximo de cada barra horizontal tiene 15 caracteres, pero si se desea más precisión puede aumentarse, ya que es troceado por la orden MID$ de la línea 180 a una longitud que se calcula por la línea 160. Esta misma línea prevé que si hay pocos caracteres siempre quede uno de ellos en el histograma.

Al pie del resultado figura un generador aleatorio que es demasiado sencillo. Con determinadas *semillas* da solamente resultados en cuatro clases, pero influye la exactitud que tiene cada computador, puesto que los números al azar generados de forma tan sencilla se denominan *seudoaleatorios* debido a que siempre existe un número a partir del cual se vuelve a repetir la secuencia de números generados, aunque si el generador es bueno esto ocurre después de varios millones de procesos.

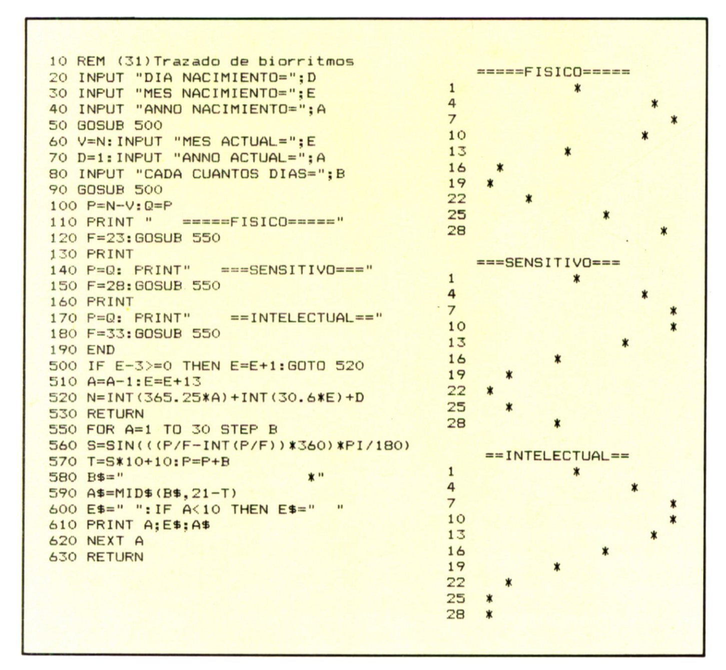
Otra posibilidad de las órdenes normales de BASIC es la

```
10 REM (30)Sinusoide
20 INPUT "INCREMENTO EN GRADOS=";A
30 B$="...................*"
40 C$=" ":D$="  ":E$="   "
50 REM Valor del seno de A
60 S=10*SIN(G*PI/180)+10
70 A$=MID$(B$,21-S)
80 REM Resultado grafico
90 IF G<10 THEN F$=E$:GOTO 120
100 IF G<100 THEN F$=D$:GOTO 120
110 IF G<1000 THEN F$=C$
120 PRINT G;F$;A$
130 G=G+A:GOTO 50
140 END


0     .........*
20    .............*
40    ................*
60    ..................*
80    ...................*
100   ...................*
120   ..................*
140   ................*
160   .............*
180   ..........*
200   ......*
220   ...*
240   .*
260   *
280   *
300   .*
320   ...*
340   ......*
360   .........*
```

Figura 17. Listado del programa n.º 30.

de obtener gráficas de funciones, tal como puede verse en el programa n.º 30 de la figura 17. Se trata de representar la gráfica de la función seno de un ángulo y entonces se presenta la cuestión de trabajar con grados sexagesimales, si el computador trabaja solamente con radianes. Se soluciona esto en la línea 60 y, como se ha elegido una longitud de la cadena de caracteres de 20 puntos, para que no queden



```
10 REM (31)Trazado de biorritmos
20 INPUT "DIA NACIMIENTO=";D
30 INPUT "MES NACIMIENTO=";E
40 INPUT "ANNO NACIMIENTO=";A
50 GOSUB 500
60 V=N:INPUT "MES ACTUAL=";E
70 D=1:INPUT "ANNO ACTUAL=";A
80 INPUT "CADA CUANTOS DIAS=";B
90 GOSUB 500
100 P=N-V:Q=P
110 PRINT "   =====FISICO====="
120 F=23:GOSUB 550
130 PRINT
140 P=Q: PRINT"   ===SENSITIVO==="
150 F=28:GOSUB 550
160 PRINT
170 P=Q: PRINT"    ==INTELECTUAL=="
180 F=33:GOSUB 550
190 END
500 IF E-3>=0 THEN E=E+1:GOTO 520
510 A=A-1:E=E+13
520 N=INT(365.25*A)+INT(30.6*E)+D
530 RETURN
550 FOR A=1 TO 30 STEP B
560 S=SIN(((P/F-INT(P/F))*360)*PI/180)
570 T=S*10+10:P=P+B
580 B$="                    *"
590 A$=MID$(B$,21-T)
600 E$=" ":IF A<10 THEN E$="  "
610 PRINT A;E$;A$
620 NEXT A
630 RETURN
```

```
   =====FISICO=====
1            *
4                    *
7                      *
10                  *
13          *
16   *
19  *
22      *
25              *
28                    *

   ===SENSITIVO===
1            *
4                   *
7                      *
10                     *
13                *
16         *
19    *
22  *
25    *
28         *

    ==INTELECTUAL==
1            *
4                  *
7                      *
10                     *
13                   *
16              *
19         *
22    *
25  *
28  *
```

Figura 18. Listado del programa n.º 31.

valores negativos se multiplica el resultado del seno por diez y se le suma también diez valores.

La cadena de caracteres puede verse en la línea 30 y se recortan a los adecuados por la línea 70, empleando el valor 21 – S para que no aparezca el valor cero, que no es admitido por la orden MID$. Como la impresión del resultado es de forma aditiva, tal como consta en la línea 120, resolvemos el problema de que los grados tengan una, dos o tres cifras poniendo tres tipos de separadores que se eligen por las líneas 90, 100 y 110. Este sistema tan sencillo tiene el

inconveniente de que los grados se han de tomar con valores enteros, lo cual condiciona el incremento de la línea 20 en la variable A. De la misma forma que se ha indicado en el programa anterior, se tendría más precisión aumentando el número de caracteres de la cadena B$, en la cual se pueden sustituir los puntos por espacios en blanco, tal como veremos en el programa siguiente.

Una aplicación de este tipo de gráficos es el *trazado de biorritmos* por el computador, que se muestra en el programa n.º 31 de la figura 18. Se ha preferido presentar el biorritmo de un mes, poniendo el título de cada ritmo en el encabezamiento y con un número de días variable que se elige al principio del programa.

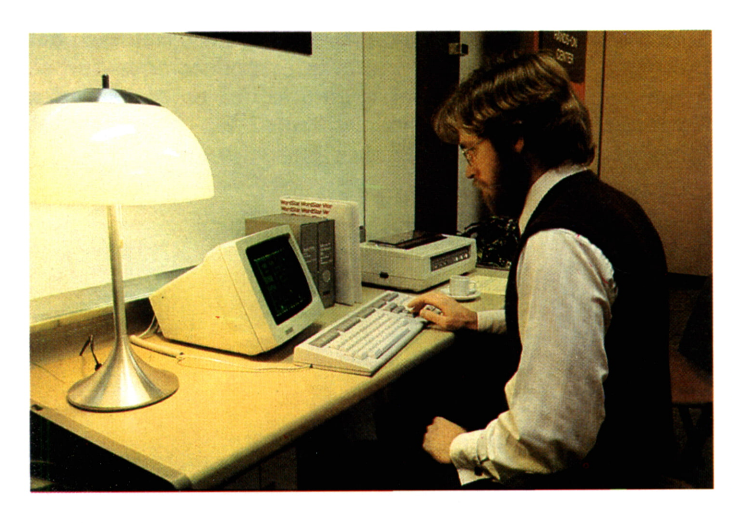
Hay numerosas obras que tratan de la forma de interpretar los ritmos cíclicos que observamos en la naturaleza. Vale la pena estudiarlos, al menos para ver si se ponen a punto por ritmos siderales o ritmos más exactos, como ocurre con el ritmo anual de la floración de las plantas, de la caída de las hojas, etc. Según estudios de Sigmund Freud, Wilhelm Fliess y Herman Swoboda, en la especie humana hay tres ritmos, uno de 23 días que controla el vigor físico, otro de 28 días que controla todo lo relacionado con lo sensitivo, la creatividad, el amor, el optimismo, y un tercer ciclo rítmico de 33 días que rige el aspecto intelectual, la capacidad de asimilación, la rapidez mental, etc. Dejando aparte la forma de interpretar estos ciclos, veamos cómo los representa el programa en la pantalla.

Como los ritmos se representan de forma independiente para tener una mayor sencillez en el programa, primero pregunta por los datos de la fecha de nacimiento y a continuación por la subrutina 500 se calculan los días pasados, teniendo en cuenta los años bisiestos.

A continuación hay que entrar el mes y año actual y la misma subrutina 500 calcula otra vez los días pasados, de manera que con los anteriores y por la línea 100 se hallan los días que han pasado desde el nacimiento, que se asignan a la variable P.

Como este valor va a utilizarse y se incrementará dentro de los valores de cada mes, se asigna a la variable *Q* que es quien los va a tener de forma permanente. Se imprime el encabezamiento del ritmo FISICO y se va a la subrutina de la línea 550 para imprimir el gráfico del mes, teniendo en cuenta que el ciclo es de 23 días en la línea 120.

Por la subrutina de la línea 550 se imprime la parte gráfica del ciclo, a través del valor del seno, en este caso también se pasan los radianes a grados sexagesimales. Se utiliza la función seno porque permite una alternancia de valores entre 1 y −1, pero podría ser representado el ritmo con líneas rectas aunque el programa sería más complicado.



El trabajo que puede realizar un computador deja abierta la posibilidad de investigación y elaboración de programas por parte del usuario.

En la línea 570 se calcula el trozo de cadena de caracteres y se incrementa el valor de los días. Esta cadena tiene como en el programa anterior 19 espacios en blanco y al final un asterisco y se recorta en la línea 590 por la orden MID$, de la misma forma en que se ha hecho en el programa anterior. Se imprime el trozo de cadena de caracteres a continuación, y se pasa a un nuevo ritmo si ya ha terminado el bucle repetitivo de la línea 620.

Cambiando los días elegidos en la línea 80 se logra un mejor aspecto, pero en el ejemplo representado se han elegido cada tres por cuestiones de espacio. Para comprobar la exactitud se parte de una fecha de nacimiento como la 1-1-1983 y como fecha actual 1-1983, con lo que los tres ciclos empiezan en el origen y se ve la repetición cada 23, 28 y 33 días.

Si bien se han representado separadamente los tres ritmos para una fácil comprensión del programa, caso de que se

desease superponerlos tendría que hacerse con las iniciales F, S, I, en vez del asterisco. Entonces el programa calcularía tres veces el trozo de cadena de caracteres (una para cada ritmo) y tendría que ordenar sus valores de menor a mayor. Una vez ordenados se hallarían las diferencias del 2.º con el 1.º y del 3.º con el 2.º, diferencias que serían trozos de cadena de caracteres con su letra al final, los cuales se imprimirían sumándolos. El resultado queda con un mayor atractivo, pero se aumenta bastante la dificultad del programa ya que aparecen nuevas complicaciones que se tienen que resolver con el algoritmo del programa, tales como la forma de imprimir cuando dos o tres ciclos coinciden y hay que decidir qué letra se imprime.



Las tabletas gráficas permiten una fácil entrada de los datos para la programación de complejos diagramas y figuras.

Además de las representaciones gráficas en baja resolución, los computadores personales pueden realizar los gráficos con mejor precisión de acuerdo con los puntos que se usen en la pantalla.

Es normal encontrar en los computadores personales pantallas con 280 × 190 puntos que en total son 53.200, es decir: unos 50.000 puntos que deben guardarse en la memoria del computador, con lo que éste ha de tener al menos 16 Kbytes de memoria. Con un computador así

puede trazarse la curva cerrada de Waclaw Sierpinski hasta el grado 4.º. Para que se vea el detalle de su forma, cuando se traza el grado 5.º hace falta un computador de 550 × 280

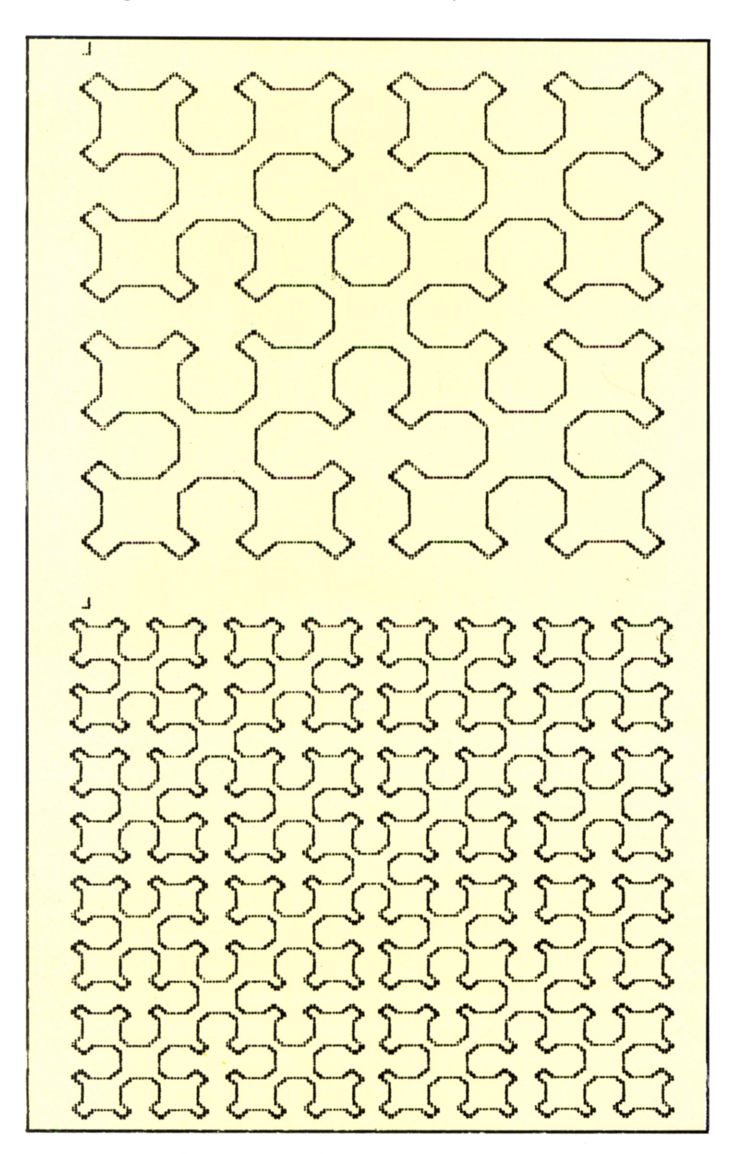
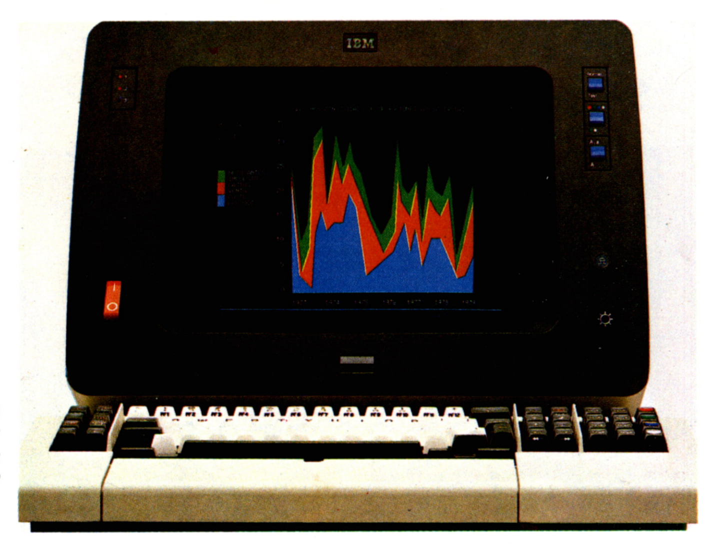


Figura 21. Curva de Walaw Sierpinski de orden 3 y 4.

puntos, que en total son 154.000 puntos. Ello indica que si se desea una mejor definición hay que aumentar la memoria del computador mucho más, hasta alcanzar la que tienen los computadores que se dedican a trabajos especializados de grafismo electrónico para efectos especiales en TV o en publicidad, memoria capaz para los cientos de miles de puntos que hay que utilizar. Véase en la figura 21 la curva de Sierpinski.



La posibilidad de programar gráficos en color proporciona un gran aliciente a la tarea de programación.

A la falta de estandarización de los puntos de la pantalla en los computadores personales se añade el hecho de que tienen órdenes para gráficos con distintos nombres, así en unos casos posiciona un punto en la pantalla con órdenes como PLOT, CURSET, PSET, etc. mientras que trazan una línea recta con DRAW, HPLOT, LINE, etc.

Estas órdenes unas veces se refieren a direcciones numéricas de los puntos de la pantalla, las cuales son absolutas, mientras que otras veces se trata de direcciones relativas, es decir, que la orden viene acompañada de una cifra que en unos computadores se refiere a una posición exacta de la pantalla, pero en otros es un incremento relativo al último punto trazado.

A todo ello se añade el hecho de que hay computadores cuyas coordenadas *X, Y,* tienen el cero en el ángulo inferior izquierdo, mientras que otros lo tienen en el ángulo superior izquierdo.



Conjunto de computador personal Apricot con sus periféricos: pantalla, floppy disc, programas e impresora a color. (Cortesía: DSE-Apricot).

Debido a todo ello, para programar las representaciones gráficas con alta resolución de la pantalla, se han reducido las órdenes a dos: una que posiciona un punto en la pantalla y la otra traza una línea recta. El hecho de la posición del cero de las coordenadas de la pantalla se resuelve restando los valores de *Y*, cuando el cero está en el ángulo superior izquierdo, del valor de puntos máximo en dirección vertical.

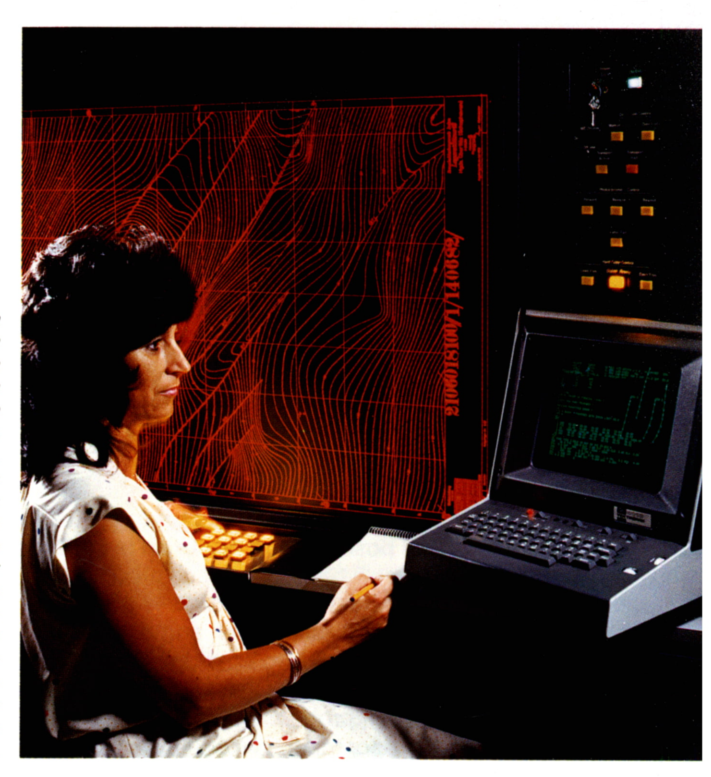
Una vez hecha esta simplificación, que se traduce en un alargamiento del listado del programa, los computadores pueden reducirse a tres grupos distintos:

Grupo A: computadores que trazan una recta desde las coordenadas absolutas X_0, Y_0 hasta las finales absolutas X_1, Y_1. Entre ellos tenemos por ejemplo la marca APPLE que utiliza la orden HPLOT X_0, Y_0 TO X_1, Y_1.

Grupo B: computadores que trazan una recta desde las coordenadas que tenía la última vez hasta las X_1, Y_1, las cuales tienen que guardarse en su memoria para el próximo trazado. Entre ellos tenemos, por ejemplo, el computador HP-86 que utiliza la orden DRAW X_1, Y_1 para trazar una

recta desde el último punto al nuevo indicado por las coordenadas X_1, Y_1.

Grupo C: computadores que trazan una recta desde las coordenadas del trazado anterior expresado en forma de puntos de la pantalla en valor entero, hasta las relativas X_1, Y_1 que significan un incremento de puntos de la pantalla en valores enteros. Entre ellos tenemos, por ejemplo, las marcas



Un área de aplicación poco conocida de la informática es la relativa a la cartografía. La armoniosa simbiosis de tres tecnologías: láser, fotoquímica y óptica, conforman este ingenioso sistema de visualización cartográfica de alta resolución que permite digitalizar, almacenar y presentar contornos de mapas en un tiempo quince veces más rápido que los sistemas manuales, al tiempo que simplifica la realización de operaciones cartográficas complejas.

ORIC y SPECTRUM, que utilizan el comando DRAW X_1, Y_1, siendo estos valores puntos de la pantalla en forma entera, y si hay decimales el computador realiza un redondeo para terminar el trazado de una recta. Debido a ello, si se utiliza un incremento en el trazado que tenga parte decimal, no se logra exactitud y una curva puede no pasar por el valor cero, tal como estaba previsto, o no se cerrará en el punto que se esperaba.

```
10  REM   (40)TRAZADO GRAFICO  DE  UNA FUNCI
    ON.
20  TEXT : HOME : REM --SEGUN CADA ORDENADOR
    --
30 RA = 3.141592654 / 180
100  INPUT "Indicar SI o NO se usan grados:"
    ;R$
110  INPUT "Valor del origen del intervalo=X
    1 ";X1
120  INPUT "Valor del final del intervalo=X2
     ";X2
130  INPUT "Valor del incremento constante=I
    N ";IN
160 P1 = 1:P2 = 278:P3 = 1:P4 = 190
170  PRINT : PRINT "Superficie de trazado ut
    ilizada:"
180  PRINT "           V4 --------"
190  PRINT "              !       !"
200  PRINT "              !       !"
210  PRINT "           V3 --------"
220  PRINT "              V1      V2"
230  PRINT
240  INPUT "V1=";V1: INPUT "V2=";V2
250  INPUT "V3=";V3: INPUT "V4=";V4
350  IF R$ = "NO" OR R$ = "N" THEN 400
370 X1 = X1 * RA:X2 = X2 * RA:IN = IN * RA
380 V1 = V1 * RA:V2 = V2 * RA:UX = UX * RA
400  GOSUB 7000: REM --CALCULOS INICIALES--
410  HGR2 : HCOLOR= 3: REM --SEGUN CADA ORDE
    NADOR--
500  REM --TRAZADO DE LA FUNCION DE LINEA 8
    010--
510 X = X1: GOSUB 8000:F$ = "0": GOSUB 7100:
    F$ = "1"
520  FOR X = IN + X1 TO X2 STEP IN
530  GOSUB 8000: GOSUB 7100
540  NEXT X: STOP
7000  REM --CALCULOS INICIALIZACION--
7010 XA = (P2 - P1) / (V2 - V1)
7020 XB = (P1 * V2 - P2 * V1) / (V2 - V1)
7030 YA = (P4 - P3) / (V4 - V3)
7040 YB = (P3 * V4 - P4 * V3) / (V4 - V3)
7050  RETURN
7100  REM --TRAZADO DENTRO SUPERFICIE UTIL--

7120 XT = X:YT = Y
7130 XS = XB + XA * XT
7140 YS = P4 - YB - YA * YT: REM --SEGUN COO
    RDENADA Y--
7150  IF XS < 0 OR XS > P2 OR YS < 0 OR YS >
    P4 THEN F$ = "0": RETURN
7160  IF F$ = "0" THEN  HPLOT XS,YS:XP = XS:
    YP = YS:F$ = "1": RETURN
7170  IF F$ = "1" THEN  HPLOT XP,YP TO XS,YS
    :XP = XS:YP = YS: RETURN
7180  PRINT "ERROR PLUMA 0-1": STOP
8000  REM ---FUNCION A REPRESENTAR--
8010 Y =  SIN (X) / X
8020  RETURN
```

Figura 25. Listado del programa n.º 32 para el trazado de una función en alta resolución. Computador tipo A.

Para los computadores del tipo B se simplifica mucho la programación ya que no hay que guardar por el programa la posición anterior del trazado, guardándose los valores de los puntos con toda precisión. Los computadores del tipo A generalmente pueden trazar como los del grupo A o los del grupo B, y tampoco presentan dificultades en la precisión de las posiciones de los puntos.

Pero en el trazado de gráficos de alta resolución con los computadores del tipo C tienen que guardarse los valores de los puntos de la pantalla en su parte decimal en variables auxiliares XK, YK, que inicialmente se ponen a cero en la línea 7005 y luego, antes de realizar el trazado de cualquier línea, se halla el valor del incremento relativo XR = XS − XP, YR = YS − YP (líneas 7170 y 7175), tras ello se calcula el valor de la parte decimal que se va acumulando en las mismas líneas del programa con la sentencia XK = XR − INT (XR) + XK, hasta que el valor de XK es mayor que 1 en que se aumenta XR en una unidad y se le resta este valor a XK (líneas 7180 y 7185), tal como puede verse en los listados parciales de las figuras 27 y 28.

Con esta complicación se logra la precisión suficiente para que los gráficos no se distorsionen, como puede verse en los ejemplos que se acompañan.

```
7000 REM --Calculos de inicializacion--
7010 XA=(P2-P1)/(V2-V1)
7020 XB=(P1*V2-P2*V1)/(V2-V1)
7030 YA=(P4-P3)/(V4-V3)
7040 YB=(P3*V4-P4*V3)/(V4-V3)
7050 RETURN
7100 REM --Trazado dentro superficie util--
7120 XT=X @ YT=Y
7130 XS=XB+XA*XT
7140 YS=YB+YA*YT ! --Segun cada ordenador--
7150 IF XS<0 OR XS>P2 OR YS<0 OR YS>P4 THEN F$="0" @ RETURN
7160 IF F$="0" THEN MOVE XS,YS @ F$="1" @ RETURN
7170 IF F$="1" THEN DRAW XS,YS @ RETURN
7180 DISP "ERROR PLUMA 0-1" @ STOP
```

Figura 26. Listado a modificar en el programa n.º 32 para un computador tipo B.

El programa número 32 de la figura 25 permite el trazado gráfico de una función con un computador tipo A. Puede verse en la línea 7170 que realiza el trazado con la orden HPLOT XP, YP TO XS, YS y luego hace el cambio de variables para tener las nuevas XP, YP de partida para la recta a trazar.

En la línea 20 se limpia la pantalla gráfica y también la de

texto normal. En la línea 30 se prepara la conversión de radianes en grados sexagesimales, conversión que se pregunta por la pantalla en la línea 100 y se realiza la transformación de las variables en las líneas 370, 380.

A continuación, por la pantalla se piden los valores extremos del intervalo de trazado y del incremento constante. En la línea 160 se determinan los valores de los puntos que tiene la pantalla, valores que se asignan a las variables *P* con un punto menos para evitar llegar a los bordes de la misma, lo cual daría lugar a un mensaje de error que detendría el trazado. Estos valores se refieren a una pantalla que tenga 280 × 192 puntos.

```
220 P1=1:P2=238:P3=1:P4=198
460 HIRES:PAPER 7:INK 0
7000 REM---INICIALIZACION---
7005 XK = 0:YK = 0
7010 XA=(P2-P1)/(V2-V1)
7020 XB=(P1*V2-P2*V1)/(V2-V1)
7030 YA=(P4-P3)/(V4-V3)
7040 YB=(P3*V4-P4*V3)/(V4-V3)
7050 RETURN
7100 REM---TRAZ.DENTRO SUPERIFICIE UTIL---
7120 XT=X:YT=Y
7130 XS=XB+XA*XT
7140 YS=P4-YB-YA*YT
7145 IF XS<0 OR XS>P2 OR  YS<0 OR YS>P4 THEN F$="0":RETURN
7150 IF F$="0" THEN CURSET XS,YS,1:XP=XS:YP=YS:F$="1":RETURN
7160 IF F$ <  >"1" THEN PRINT "ERROR 0-1 PLUMA": STOP
7170 XR = XS - XP:XK = XR - INT (XR) + XK
7175 YR = YS - YP:YK = YR - INT (YR) + YK
7180 IF XK > 1 THEN XR = XR + 1:XK = XK - 1
7185 IF YK > 1 THEN YR = YR + 1:YK = YK - 1
7190 DRAW INT (XR), INT (YR),1:XP = XS:YP = YS: RETURN
```

Figura 27. Modificaciones a efectuar en el programa n.º 32 para un computador tipo C modelo Oric I.

Por la pantalla aparece a continuación el recuadro de la superficie de trazado por las variables V_1, V_2 (horizontal) y V_3, V_4 (vertical). Los primeros son los valores que se han elegido en el intervalo en las líneas 110 y 120, pero V_3 y V_4 dependen de las ordenadas que tenga el trazado según la función a representar, valores que no podrán presentar mensaje de error por trazarse fuera de la pantalla por el filtro de la línea 7150.

Este filtro y los cálculos para el trazado están en función de los puntos de la pantalla entrados anteriormente con las variables de P_1 a P_4.

El trazado se realiza posicionando el punto de partida por la línea 510, que envía a la subrutina de la línea 8000, la cual contiene la función a representar, y la de la línea 7100 que posiciona el punto de partida si hay el valor 0 en la variable

F$, y traza una línea si esta variable tiene el valor 1. Esto puede comprobarse en las líneas 7160 y 7170.

El punto de partida citado antes se establece con los cálculos de las líneas 7000 a 7050 que hallan el posicionamiento al iniciarse el trazado en la línea 400. Puede verse que en la línea siguiente figuran las órdenes que tiene cada computador para pasar a la pantalla de alta definición, mientras que el trazado se realiza por el bucle de la línea 520 en función del incremento constante IN.

En la utilización de este programa hay que introducir la función a representar escribiendo la línea 8010 previamente a pulsar RUN y luego contestar los mensajes que aparecen por la pantalla. Puede verse en las modificaciones del listado que aparecen en la figura 26 la simplificación de la programación que presenta un cimputador del tipo B (HP-86) con los mismos números de programa que en la figura anterior.

En la figura 27 pueden verse las modificaciones que ha de sufrir el programa n.º 32 para un computador del tipo C (ORIC) que tiene el origen de coordenadas en el ángulo superior izquierda (línea 7140), por ello hay que restar del valor de P_4 los demás valores de las variables. Se tienen que iniciar en la línea 7005 los valores de XK, YK que contendrán la parte decimal de los puntos de la pantalla, la cual se irá agregando a las variables XR, YR a medida que sobrepase el valor de la unidad.

Con ello se logra la precisión necesaria para que el trazado resulte con toda perfección.

Para otro computador del tipo C, pero que tiene el origen de coordenadas distinto (SPECTRUM), en la figura 28 pueden verse las modificaciones que hay que introducir en el listado del programa n.º 32. La línea 7140 ya no tiene la diferencia con el valor total de P_4 como con el computador anterior y las demás variables son las citadas anteriormente, con la finalidad de acumular las partes decimales de las direcciones de los puntos de la pantalla. Este computador utiliza letras minúsculas para las variables, lo cual no es corriente en los computadores personales.

Hay varios ejemplos de la utilización del programa n.º 32 que permiten ver la comodidad que representa el utilizar los gráficos en alta resolución, aunque solamente sea con pantallas de unos 50.000 puntos que es lo corriente en los computadores personales, pero se presenta en el programa

n.º 33 de la figura 29 otra aplicación del trazado en alta resolución, la obtención de la pirámide de edades de una población determinada.

```
 220 LET p1=1: LET p2=254: LET p
3=1: LET p4=174

 455 CLS
 460 PAPER 7: INK 0: REM --Segun
 cada ordenador--

7000 REM --Inicializacion--
7005 LET xk=0: LET yk=0
7010 LET xa=(p2-p1)/(v2-v1)
7020 LET xb=(p1*v2-p2*v1)/(v2-v1
)
7030 LET ya=(p4-p3)/(v4-v3)
7040 LET yb=(p3*v4-p4*v3)/(v4-v3
)
7050 RETURN

7100 REM --Trazado en superf.util.
7120 LET xt=x: LET yt=y
7130 LET xs=xb+xa*xt
7140 LET ys=yb+ya*yt
7145 IF xs<0 OR xs>p2 OR ys<0 OR
 ys>p4 THEN LET F$="0": RETURN
7150 IF F$="0" THEN PLOT xs,ys:
LET xp=xs: LET yp=ys: LET F$="1"
: RETURN
7160 IF F$<>"1" THEN PRINT "erro
r plu": STOP
7170 LET xr=xs-xp: LET xk=xr-INT
 (xr)+xk
7175 LET yr=ys-yp: LET yk=yr-INT
 (yr)+yk
7180 IF xk>1 THEN LET xr=xr+1: L
ET xk=xk-1
7185 IF yk>1 THEN LET yr=yr+1: L
ET yk=yk-1
7190 DRAW INT (xr),INT (yr): LET
 xp=xs: LET yp=ys: RETURN
```

Figura 28. Modificaciones a efectuar en el programa n.º 32 para un computador tipo C (Spectrum).

Las pirámides de edades sirven para estudiar en demografía el estado de una población, separando el número de hombres del de mujeres y en cada uno de ellos se representa

```
10  REM  (41)PIRAMIDE EDADES BARCELONA-80.
20  DIM H(110): DIM M(110)
30  TEXT : HOME : REM  --SEGUN CADA ORDENADOR--
100  REM ---ENTRADA DE DATOS---
105 J = 1
110  READ H: IF H = 0 THEN HB = J - 1: GOTO 160
120 J = J + 1:H(J) = H: GOTO 110
160 J = 1
170  READ M: IF M = 0 THEN MB = J - 1: GOTO 210
180 J = J + 1:M(J) = M: GOTO 170
210  PRINT : PRINT "Modificar las variables P1,P2
     ,P3,P4 "
220  PRINT : PRINT "si la pantalla no es de 280x1
     92 puntos."
230 P1 = 1:P2 = 278:P3 = 1:P4 = 190
240 B = 300
245 C = B * (P2 - P1) / 6 / P2
250 V1 =  - B:V2 = B + C:V3 = 0:V4 = 95
260 UX = 50: REM  --DIVIS.CADA 5000 HOMBRES--
270 UY = 10: REM --DIVIS.CADA 10 ANNOS--
280  HGR2 : HCOLOR= 3: GOSUB 7000
290  GOSUB 7200: GOSUB 7400: REM --TRAZ.EJES--
300  REM --TRAZADO BARRAS DE HOMBRES--
310 X = 0:Y = 0:F$ = "0": GOSUB 7100
320  FOR J = 1 TO HB
330 X =  - H(J):F$ = "1": GOSUB 7100
340 Y = J: GOSUB 7100
350 X = 0: GOSUB 7100
360  NEXT J
400  REM --TRAZADO BARRAS DE MUJERES--
410 X = C:Y = 0:F$ = "0": GOSUB 7100
420  FOR J = 1 TO MB
430 Y = J: GOSUB 7100
440 X = M(J) + C: GOSUB 7100
450 Y = J - 1: GOSUB 7100
460 X = C: GOSUB 7100
470  NEXT J: STOP
7000  REM  ---CALCULOS INICIALIZACION---
7010 XA = (P2 - P1) / (V2 - V1)
7020 XB = (P1 * V2 - P2 * V1) / (V2 - V1)
7030 YA = (P4 - P3) / (V4 - V3)
7040 YB = (P3 * V4 - P4 * V3) / (V4 - V3)
7050  RETURN
7100  REM  ---TRAZADO DENTRO SUPERFICIE UTIL---
7120 XT = X:YT = Y
7130 XS = XB + XA * XT
7140 YS = P4 - YB - YA * YT: REM --SEGUN CADA ORD
     ENADOR
7145  IF XS < 0 OR XS > P2 OR YS < 0 OR YS > P4 THEN
     F$ = "0": RETURN
7150  IF F$ = "0" THEN  HPLOT XS,YS:XP = XS:YP =
     YS:F$ = "1": RETURN
7160  IF F$ = "1" THEN  HPLOT XP,YP TO XS,YS:XP =
     XS:YP = YS: RETURN
7170  PRINT "ERROR 1-0 PLUMA": STOP
7200  REM  ---TRAZADO DE EJES---
7210 EX = V1:EY = V3: REM  --COMPRUEBA PASAN POF
     SUPERFICIE UTIL--
7220  IF V1 < 0 AND V2 > 0 THEN EX = 0
7230  IF V3 < 0 AND V4 > 0 THEN EY = 0
7240 XT = V1:YT = EY
7250 F$ = "0": GOSUB 7130
7260 XT = V2:F$ = "1": GOSUB 7130
7270 XT = EX:YT = V3
7280 F$ = "0": GOSUB 7130
7290 YT = V4:F$ = "1": GOSUB 7130
7300  RETURN
```

Figura 29. Listado del programa n.º 33, distribuido entre las páginas 38 y 39.

```
7400  REM ---TAZADO GRADUACIONES EJES--
7410 GX = ((EX - V1) / UX -  INT ((EX - V1) / UX)
     ) * UX
7420 GY = ((EY - V3) / UY -  INT ((EY - V3) / UY)
     ) * UY
7425  REM --DIVISIONES EN EJE X--
7430  FOR XT = GX + V1 TO V2 STEP UX
7440 YT = EY:F$ = "0": GOSUB 7130
7450 YT = EY + (V4 - V3) * 4 / (P4 - P3):F$ = "1"
     : GOSUB 7130
7460  NEXT XT
7470  REM --DIVISIONES EN EJE Y--
7480  FOR YT = GY + V3 TO V4 STEP UY
7490 XT = EX:F$ = "0": GOSUB 7130
7500 XT = EX + (V2 - V1) * 4 / (P2 - P1):F$ = "1"
     : GOSUB 7130
7510  NEXT YT
7520  RETURN

8000  REM --DATOS POBLACION DE HOMBRES--
8010  DATA  207,219,233,247,272,277,275,278,280,2
     78
8020  DATA  272,268,283,290,287,295,278,276,270,2
     69
8030  DATA  266,264,255,241,233,222,217,220,215,2
     19
8040  DATA  231,238,242,220,239,228,226,199,189,2
     39
8050  DATA  160,167,211,226,224,226,232,236,233,2
     37
8060  DATA  225,226,220,223,216,215,211,203,198,1
     76
8070  DATA  155,147,141,139,136,138,131,130,120,1
     15
8080  DATA  108,104,95,90,79,72,68,61,53,49
8090  DATA  41,33,28,25,21,16,12,9,7,5,6,3,2,1,1,
     0
8200  REM --DATOS POBLACION DE MUJERES--
8210  DATA  199,205,224,233,250,258,261,262,263,2
     61
8220  DATA  263,257,272,273,275,280,275,265,260,2
     59
8230  DATA  260,254,248,238,234,228,233,235,231,2
     36
8240  DATA  241,255,254,229,251,242,238,209,196,2
     42
8250  DATA  170,176,221,246,245,237,253,253,257,2
     67
8260  DATA  254,255,253,258,251,252,244,244,231,2
     29
8270  DATA  207,202,193,186,189,194,187,185,178,1
     76
8280  DATA  172,164,151,151,140,130,123,116,101,1
     05
8290  DATA  81,69,60,55,48,39,31,26,19,16,16,11,6
     ,5,1,0
```

por una línea horizontal el número de individuos que hay en cada año. Puede verse el resultado en la figura 30 del censo de 1980 y para la ciudad de Barcelona.

Este es un caso en el que se aprecia la ventaja de tener los datos traducidos en forma gráfica porque permite una mejor comprensión del estado de la población, como la disminución de la natalidad actual, la influencia de la baja de natalidad por la guerra del 36, el aumento de la población de mujeres de edad superior a los 60 años sobre la equivalente de hombres, etc.

El programa n.º 33 de la figura 29 realiza el trazado para un computador del tipo A, tal como puede verse en las líneas 7145 para el filtro que elimina las posiciones del exterior de la pantalla, línea 7150 que posiciona el principio de cada trazado de las líneas con la orden HPLOT, y en la línea 7160 hay el trazado de cada línea con la orden HPLOT XP, YP TO XS, YS.

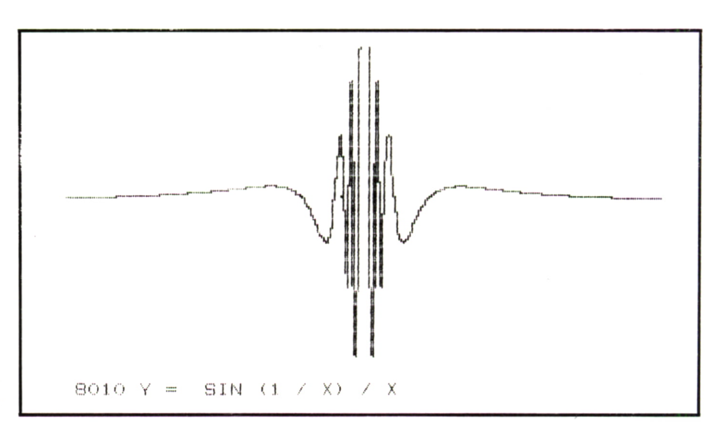


Figura 30. Trazado de la función y=sin (1/x)/x

El programa presenta suficientes comentarios con REM, que permiten seguir el procedimiento seguido en la programación, la cual en esencia consiste en el trazado de pequeños rectángulos horizontales, en función del número de individuos que tiene cada edad. Estos datos están incluidos en las líneas de la 8000 a la 8290 con la orden DATA que se lee con READ de las líneas 110 y 170. Para que se interrumpa la lectura de datos el último tiene que ser cero, lo cual se pregunta cada vez por las líneas citadas 110 y 170 con la orden IF.

Pueden apreciarse las ventajas del trazado en alta

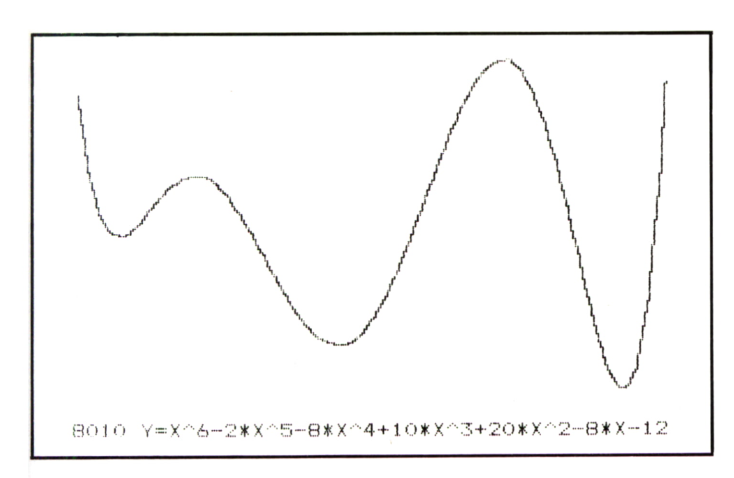


Figura 31. Trazado del polinomio $y=x^6-2x^5-8x^4+10x^3+20x^2-8x-12$.

resolución, pues si observamos las cifras de la figura 29 que corresponden a los datos del programa n.º 33, no se podrían ver inmediatamente todas las relaciones que luego aparecen en la pirámide de edades de la población de la figura 32.

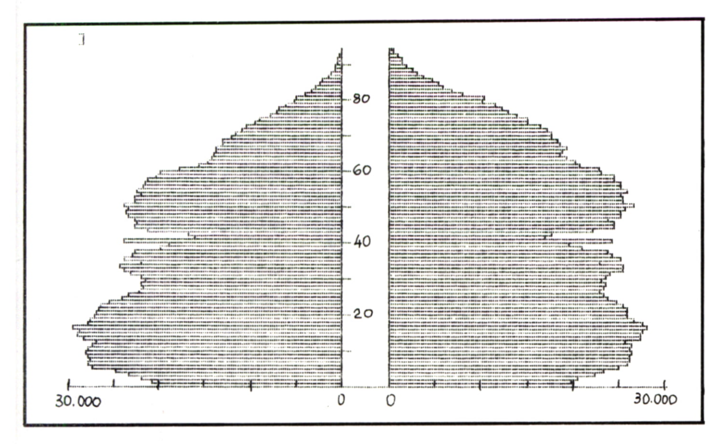


Figura 32. Pirámide de edades de la población de Barcelona según el censo de 1980-81. A la izquierda se presenta la población de hombres, y a la derecha la de mujeres.

No se puede terminar el apartado de representaciones gráficas sin tratar el tema de los gráficos con alta resolución en la pantalla. Este tema depende de las órdenes que tiene cada computador, así como de si tiene color.

Las órdenes de alta resolución suelen permitir la representación de un punto en la pantalla dadas sus coordenadas X, Y, a las que se añade el color como una cifra más. Los nombres son variados, como PLOT, PEN, CURSET, etc. y generalmente los computadores tienen una orden que permite pasar de la resolución de texto a la alta, como por ejemplo HIRES, que son las iniciales de «high resolution».

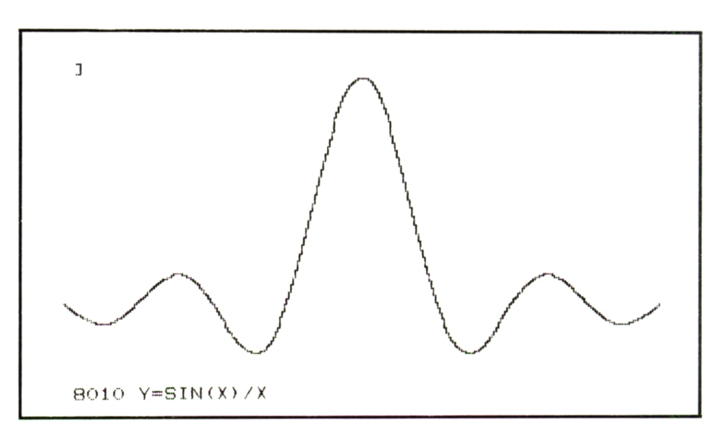


Figura 33. Trazado de la función y=sin (x)/x.

Si bien el trazado de gráficos de funciones no ofrece dificultad, si no es la de salirse de los límites de la pantalla (extremo que puede evitarse con una condición IF), hay que tener en cuenta que la pantalla no tiene valores negativos. Pero más dificultad ofrece la presentación por el computador de figuras tridimensionales, ya sean de revolución o no.

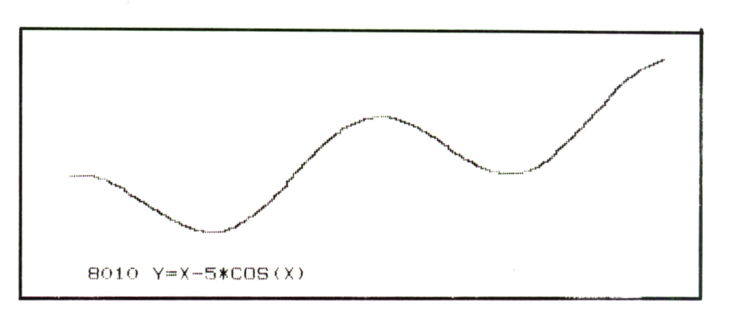


Figura 34. Trazado de la función y=x−5 cos (x).

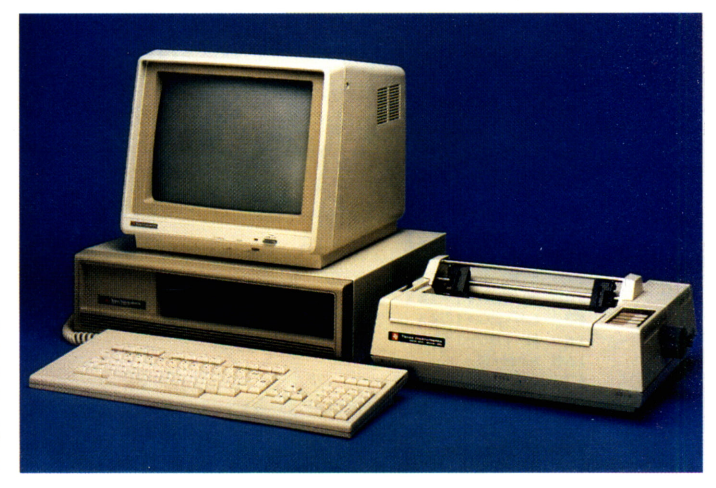
Un caso de estas representaciones lo tenemos en el programa n.º 34 de la figura 35, que consiste en la representación de una figura sencilla como un cono mediante diez circunferencias de su superficie, las cuales permiten una representación de su figura volumétrica.

```
10 REM (32)Graficos tridimensionales
20 HIRES:PAPER 7:INK 0
25 FOR U=3 TO 0 STEP-.3
27 N=N+6
30 FOR X=-U TO U STEP U/20
40 Y=SQR(1- X*X/(U*U))*(2* ABS(U)/3)
50 L=80  + 15*X
60 M=100 + 15*Y
65 K=M+N
70 CURSET L,K,1
80 NEXT X
130 FOR X=U TO -U STEP -U/20
140 Y=SQR(1-X*X/(U*U))*(2* ABS(U)/3)
150 L= 80+15*X
160 M=100-15*Y
165 K=M+N
170 CURSET L,K,1
180 NEXT X
190 NEXT U
200 END
```

Figura 35. Listado del programa n.º 34.

Estas circunferencias quedan en el papel como elipses de semiejes decrecientes, pero de la misma excentricidad. Y como la fórmula a representar es una raíz cuadrada que tiene dobles valores (positivos y negativos), se repiten dos veces las líneas del programa, no empleándose subrutinas para que el programa sea más comprensible. En la repetición se han elegido los pasos de programas que tienen el 1 delante a

partir del paso 30 y, como puede verse en los pasos 60 y 160, en uno se suma y en otro se resta para tener el valor *M* que se empleará en las líneas 70 y 170 a través de la variable *K* del paso precedente. La línea 20 es propia del computador para determinar la alta resolución de la pantalla y el tipo de color de los puntos y del fondo.



Modelo de Texas Instruments con los cuatro componentes básicos de una unidad de programación: teclado, pantalla, unidad de discos e impresora.

Las representaciones programadas de gráficos tridimensionales permiten una gran variedad de aplicaciones, dado que es fácil cambiar los parámetros si se asignan a variables y con ello se facilitan las representaciones en perspectiva de figuras complicadas, edificios, piezas de máquinas, etc. que se pueden visualizar en la pantalla desde cualquier punto de vista.

Este apartado se podría completar con las representaciones gráficas por *tablas trazadoras* o *ploters*, que suprimen los puntos al trazar líneas continuas en varios colores, posicionadas con precisión sobre el papel, con lo que se facilita la creación gráfica mediante las órdenes del lenguaje BASIC.

LA PRECISION AMPLIADA

Aunque cada computador presente pequeños errores en el

proceso de los datos, manejando convenientemente la programación se alcanza una precisión insospechada, generalmente almacenando los datos parciales en distintas variables. Entonces la precisión viene condicionada por la memoria disponible del computador, que suele ser inmensa.

Nos daremos cuenta de la importancia de este apartado si tenemos en cuenta que las cifras que podemos tener en una operación sencilla como la división periódica, normalmente están dentro de las que puede presentar en la pantalla cada variable. A veces se trata de ocho, hay veces que se llega a quince debido a que el computador trabaja con doble precisión. Esta doble precisión se logra almacenando los

```
10 REM (33)Cifras division periodica
20 INPUT "DIVIDENDO=";B
30 INPUT "DIVISOR=";C
40 PRINT "RESULTADO DE: ";B;"/";C:PRINT
50 FOR A=1 TO 8
60 E=INT(B/C)
70 B=(B-C*E)*1000
80 A$="000":B$=STR$(E)
90 C$=A$+B$:D=LEN(C$)-3
100 E$= MID$ (C$,D)
110 PRINT E$;".";
120 NEXT A
130 PRINT:GOTO 50


RESULTADO DE: 71361 / 71327

001,000.476.677.835.882.625.092.
882.078.315.364.448.245.404.965.
861.454.988.994.349.965.651.155.
943.752.015.365.850.239.039.914.
758.786.995.107.042.214.028.348.
311.298.666.704.053.163.598.637.
262.186.829.671.793.290.058.463.
134.577.369.018.744.654.899.266.
757.328.921.726.695.360.803.061.
954.098.728.391.773.101.350.119.
870.455.788.130.721.886.522.635.
187.236.249.947.425.238.689.416.
350.049.770.774.040.685.855.286.
217.000.574.817.390.329.047.906.
122.506.203.821.834.648.870.694.
127.048.663.199.069.076.226.393.
```

Figura 37. Listado del programa n.º 35.

resultados en dos variables y ello nos da pie para almacenar en más variables mediante la adecuada programación, por lo que la precisión de una operación como la división queda enormemente ampliada.

Puede verse un ejemplo de ello, en el programa n.º 35 de la figura 37, que resuelve la división por el algoritmo corriente pero almacena los resultados en ocho grupos de tres cifras cada uno, tal como puede verse en el diagrama de flujo de la figura 38.

Después de entrar los datos del dividendo y divisor el programa los imprime para que queden junto al resultado, y a continuación empieza un bucle repetitivo que imprime ocho veces el resultado de la división en grupos de tres cifras, obtenidas por el algoritmo de la división con la modificación de procesar las cifras de tres en tres por la unidad seguida de tres ceros que multiplica el resto en la línea 70. Se eligen tres cifras para que sean fáciles de leer, pero puede modificarse el programa multiplicando por la unidad seguida de más ceros hasta que los resultados obtenidos no coincidan con los que acompañan el listado del programa. Con ello se comprobará el número de cifras con que se procesan las variables, que a veces es superior a las presentadas en la pantalla.

En las líneas 80, 90 y 100 se agregan los ceros que se necesiten a la izquierda. Como aquí el resultado tiene grupos de tres cifras, en la línea 80 se asigna una cadena de caracteres a la variable A$, mientras que el resultado se transforma a cadena de caracteres y se asigna a B$. Al sumar estas variables quedará una cadena máxima de seis caracteres que irá disminuyendo si el resultado tiene ceros a la izquierda. Por LEN se halla la longitud que tiene y se recorta convenientemente en la línea 100, pasándose a imprimir con un punto de separación en la línea 110.

En este tipo de programas es necesario tener los resultados con los ceros a la izquierda, sobre todo si son todas nulas en que pierde la correcta alineación. Después del primer resultado tiene que ponerse una coma decimal, lo cual complicaría el programa, pero se haría con una bifurcación condicional que comprobaría si el resultado es mayor que cero.

Puede comprobarse que en el ejemplo que figura al pie del programa, después de calcular 382 cifras de la división no aparece el período, seguramente por tratarse de dos números cercanos en que el divisor es primo.

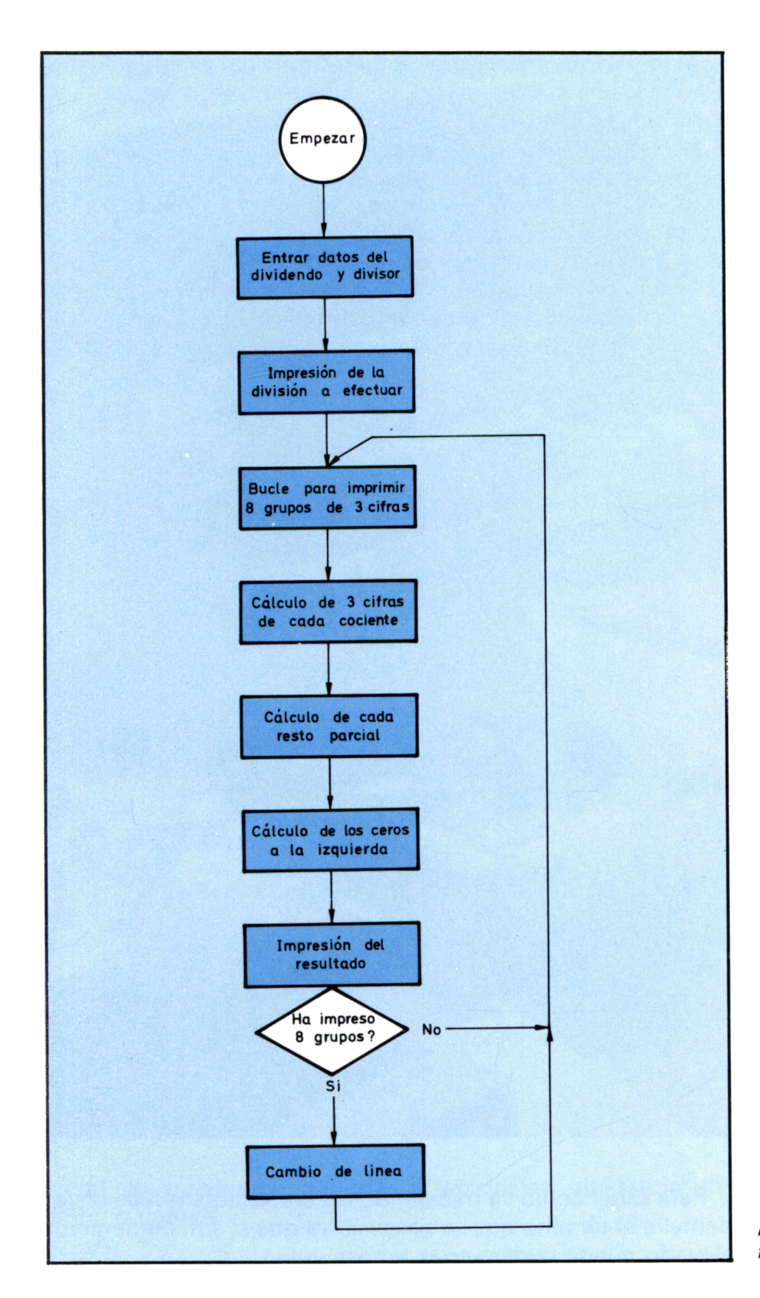
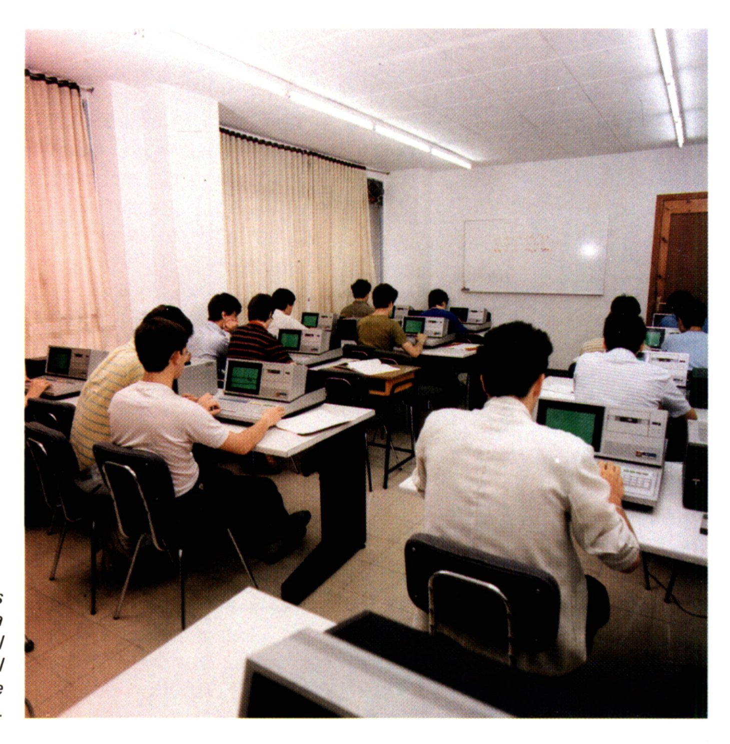


Figura 38. Diagrama de flujo del programa n.º 35.

Algo más complicado es el algoritmo y su programa para la obtención de las cifras del número «*e*», base de los logaritmos neperianos.

Para la obtención de estas cifras se recurre a una de las series matemáticas que proporcionan su valor a medida que el número de términos crece. Como nos tenemos que contentar con procesar con el programa un número de términos finito, las cifras que se pueden obtener del número «*e*» están en relación con el número de términos de la serie, que para mayor comodidad en la programación se empieza con un término máximo para procesarlos todos hasta llegar al primero.



La importancia de los computadores en la vida personal y laboral acrecienta el interés del gran público sobre este campo de la técnica.

Para saber cómo va trabajando el computador queda en la pantalla el término que se procesa, ya que si son en número elevado puede tardar varias horas.



Conjunto compacto y portátil CPDS 100 de la firma AIM.
Posee varios sistemas multiprocesadores desde 256 kbytes a 1 Mbyte en RAM, 1 Mbyte en floppy disc formateado, además de análisis lógico y las restantes funciones para desarrollar el tratamiento de información.

La serie a programar es la siguiente:

$$e = \lim_{n \to \infty} 1 + 1/1! + 1/2! + 1/3! + ... + 1/n!$$

y al tener el denominador el valor del factorial de números crecientes no puede almacenarse el resultado con exactitud en las variables numéricas, por lo que se saca factor común poniendo la suma de términos de mayor a menor, quedando como sigue:

$$....(((((1/7+1)\cdot 1/6+1)\cdot 1/5+1)\cdot 1/4+1)\cdot 1/3+1)\cdot 1/2+2= \\ =e$$

y con ello quedan suprimidos los factoriales, quedando un término general:

$$....(1/n+1)\cdot 1/(n-1)+....$$

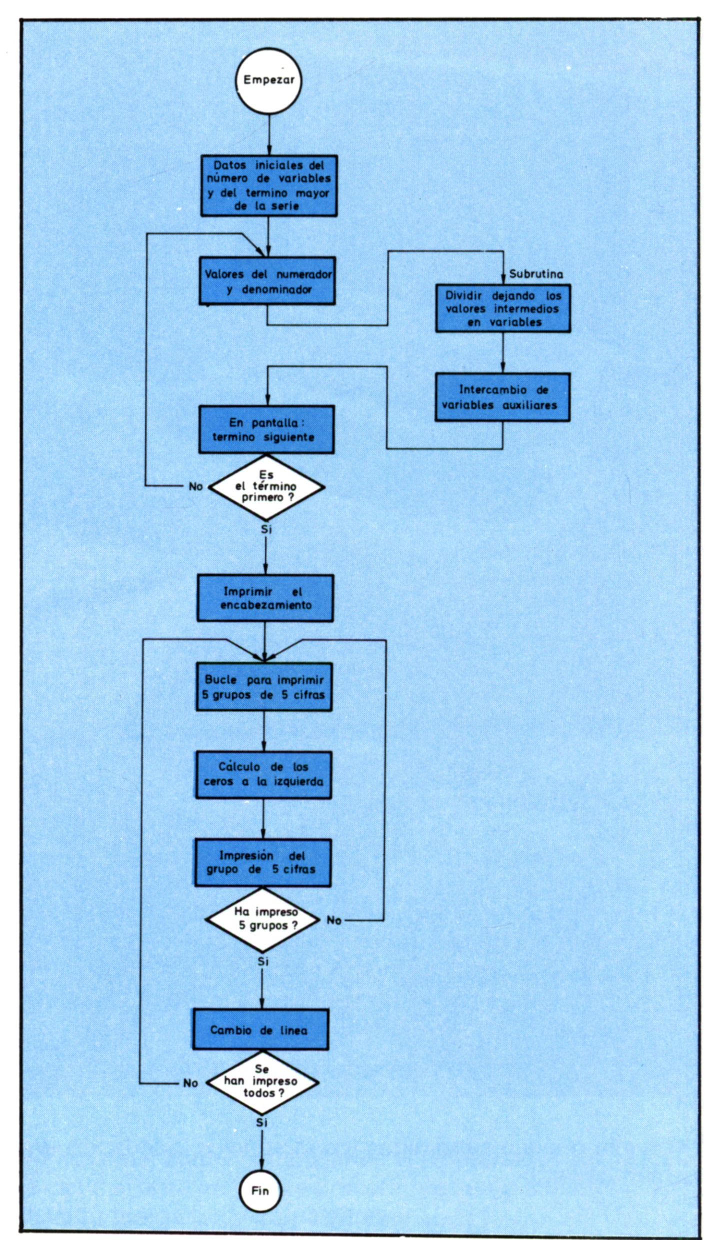


Figura 41. Diagrama de flujo del programa n.º 36.

que se programa en el diagrama de flujo de la figura 41 y en el listado de la figura 42 que corresponde al programa n.º 36.

Podemos ver que hay una subrutina que hace la división

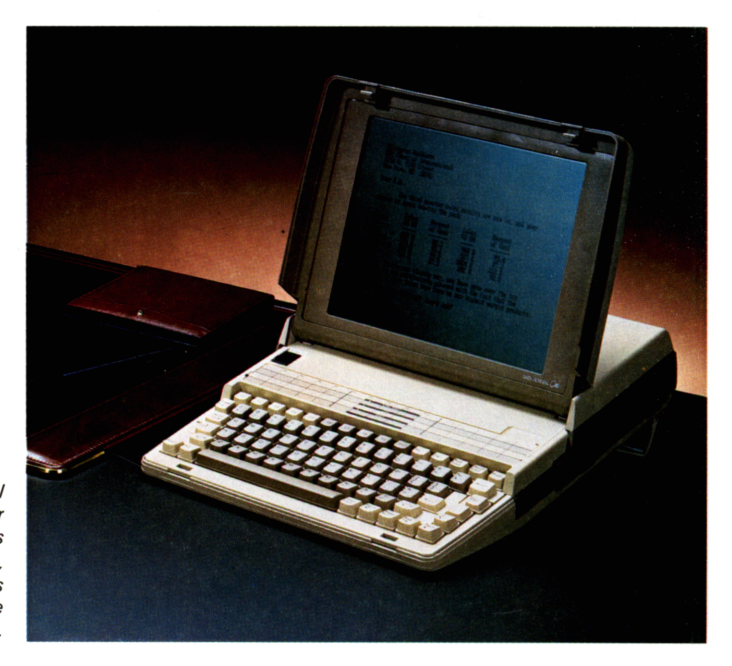
```
10 REM (34)Cifras del numero e
20 INPUT"NUMERO DE VARIABLES=";V
30 INPUT"TERMINO MAYOR DE LA SERIE";A
50 PRINT:DIM F(V+5):DIM H(V+5):W=A
60 REM Terminos de la serie
70 B=1:C=A:GOSUB 100
80 A=A-1:PRINT A:IF A=1 THEN 300
90 GOTO 70
100 REM Division de variables
110 FOR D=1 TO V
120 E=INT(B/C):F(D)=E
130 B=(B-E*C)*1E5
140 B=H(D+1)+B:NEXT D
170 REM Cambio de variables
180 FOR D=1 TO V:H(D)=F(D)
190 NEXT D
200 RETURN
300 REM Resultado impreso
310 F(1)=2
320 LPRINT"***CIFRAS DEL NUMERO e.***
330 LPRINT"TERMINO MAYOR DE LA SERIE=";W
335 LPRINT
340 FOR D=1 TO V STEP 5
350 FOR U=0 TO 4:A$="00000"
360 B$=STR$(F(D+U)):C$=A$+B$
370 Z=LEN(C$)-5:E$=MID$(C$,Z)
380 LPRINT E$;" ";
390 NEXT U
400 LPRINT:NEXT D
410 END
```

```
***CIFRAS DEL NUMERO e.***
TERMINO MAYOR DE LA SERIE=160

00002,71828 18284 59045 23536
02874 71352 66249 77572 47093
69995 95749 66967 62772 40766
30353 54759 45713 82178 52516
64274 27466 39193 20030 59921
81741 35966 29043 57290 03342
95260 59563 07381 32328 62794
34907 63233 82988 07531 95251
01901 15738 34187 93070 21540
89149 93488 41675 09244 76146
06680 82264 80016 84774 11853
74234 54424 37107 00000 00000
```

Figura 42. Listado del programa n.º 36.

de las variables necesarias para tener todos los valores exactos, a la vez que en cada nuevo cálculo se intercambian con otras auxiliares. Debido a ello, el número de variables que se maneja queda determinado al principio por la variable *V* que determina el DIM indirecto. Si el computador no permite este dimensionamiento indirecto, queda el recurso de ponerlo de forma manual en el listado del programa, añadiendo unas cuatro o cinco más para que no falten en la impresión del resultado final.



La pantalla de cristal líquido permite un menor volumen de los computadores personales, lo que les hace accesibles a cualquier lugar de trabajo.

El término máximo de la serie con el que se empieza el proceso de cálculo sirve para determinar el número de cifras exactas del resultado, tal como se ha indicado antes. Este número de cifras en el ejemplo que se acompaña es de 286 y no se han utilizado más variables porque sus valores ya no son exactos.

Para determinar el término mayor de la serie a partir de un número de cifras exactas del resultado, tiene que hallarse

previamente el número de cifras que tiene el factorial de este término. Como el factorial del número 160 tiene 285 cifras se han obtenido 286 exactas. El programa siguiente puede utilizarse para hallar las cifras del factorial de un número dado.



El interés que demuestra el gran público por los computadores personales queda reflejado en esta fotografía correspondiente a una Feria del sector.

Si observamos el programa del número «*e*» veremos que el resultado impreso de la línea 300 empieza poniendo el valor 2 a la primera variable, luego imprime el encabezamiento separado por una línea en blanco del grupo de cifras del resultado, las cuales se han impreso en cinco grupos de cinco cifras. Si el computador tiene mayor precisión pueden utilizarse más cifras en las variables, lo cual acorta el tiempo del proceso de cálculo.

Para hacer los cinco grupos de cifras se usa el bucle repetitivo de la línea 350, que tal como en el programa anterior coloca los necesarios a la izquierda del resultado. Como a la primera variable F(1) se le asigna valor aparte del bucle, podría incluírsele la coma decimal con facilidad, ya que en los grupos de cinco cifras se ha preferido separarlos

```
10 REM (35)Cifras del factorial de un num.
20 N=1:B=0
30 FOR A=N TO 1 STEP -1
40 B=B+LN(A)/LN(10)
50 NEXT A:C=INT(B)+1
60 PRINT "NUM=";N;"CIF.FACT.=";C
70 B=0:N=N*2:GOTO 30
80 END


NUM=1 CIF.FACT.=1
NUM=2 CIF.FACT.=1
NUM=4 CIF.FACT.=2
NUM=8 CIF.FACT.=5
NUM=16 CIF.FACT.=14
NUM=32 CIF.FACT.=36
NUM=64 CIF.FACT.=90
NUM=128 CIF.FACT.=216
NUM=256 CIF.FACT.=507
NUM=512 CIF.FACT.=1167
NUM=1024 CIF.FACT.=2640
NUM=2048 CIF.FACT.=5895
NUM=4096 CIF.FACT.=13020
```

Figura 45. Listado del programa n.º 37.

por un espacio en blanco. El número de cifras a imprimir depende de la variable *V* ya citada en lo referente al número de cifras que contiene cada variable auxiliar del proceso de cálculo.



Conjunto de computador Casio con sus periféricos. A la derecha se observan las unidades lectoras de los discos flexibles.

Como puede verse, este número de cifras a obtener puede ser muy elevado. Una vez conocidas las que el computador puede contener con seguridad en cada variable, se determina el número de variables y se busca un número A cuyo número de cifras de su factorial sea superior en un 5 % a las cifras que se desean tener. Este pequeño margen se pone



Computador personal de tipo profesional, provisto de pantalla giratoria para facilitar su observación desde cualquier lugar próximo. (Cortesía: DSE-Apricot).

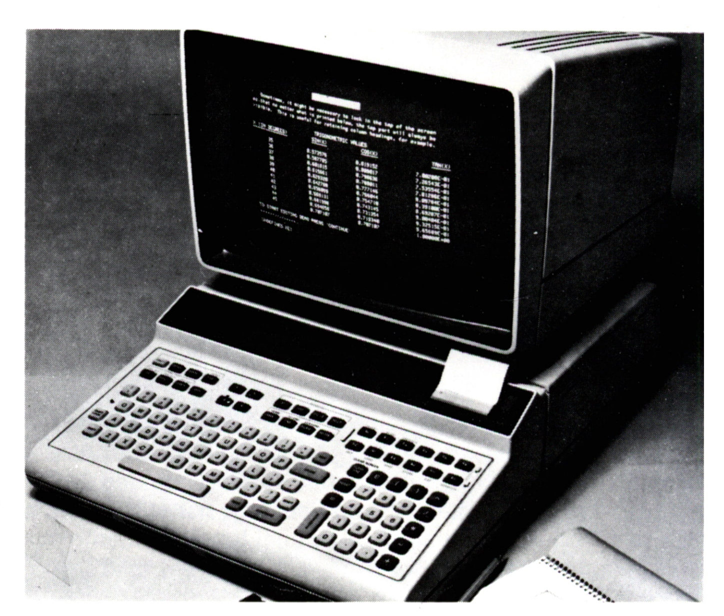
como seguridad para los últimos valores del resultado. El número *A* será el término mayor de la serie que irá decreciendo de uno en uno hasta llegar a la unidad, valor que aparece en la pantalla del computador por el PRINT de la línea 80 y que nos garantiza el normal funcionamiento del programa.

El número de variables se eleva al desear tener más cifras y,

como también es más elevado el número de operaciones a realizar por el programa, el tiempo del proceso crece en progresión geométrica. Así, en el ejemplo que se incluye como el número de variables ha sido de 58 y el término inicial de 160, el programa ha realizado 9.280 veces el cálculo y asignación a variables, de manera que al final el resultado ha quedado sin ningún error puesto que ha manejado cinco cifras por variable de las ocho que puede utilizar el computador, debido a que en el programa no hay restas con decimales que siempre introducen errores. Con ello el tiempo empleado no ha llegado a los 10 minutos.

Como auxiliar de este interesante caso de precisión ampliada, el programa n.º 37 de la figura 45 permite calcular las cifras del factorial de cualquier número. Por la variable *N* de la línea 70 se logran los resultados en progresión geométrica, pero sería fácil modificarlo para que con INPUT N en la línea 20 y suprimiendo la línea 70, nos diese las cifras del factorial de cualquier número.

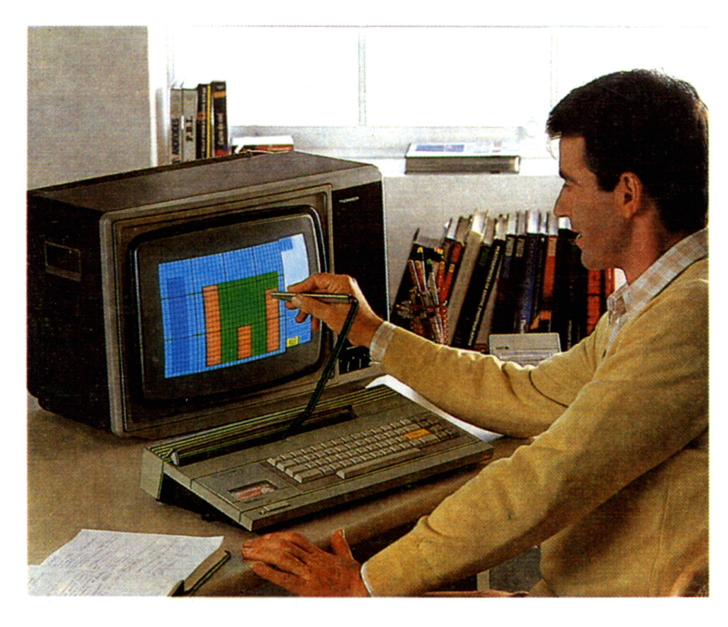
Puede utilizarse para el programa anterior y el cálculo se realiza empleando el logaritmo decimal una vez convertido del neperiano, si esta función no existe en el computador tal



Para las aplicaciones comerciales y de gestión, el teclado de cifras de la derecha resulta de gran utilidad.

como puede verse en la línea 40. El número de cifras se va sumando a la variable B que se asigna a C y da el resultado final.

El tema de la precisión en el proceso de cálculo por los computadores y empleando un lenguaje de alto nivel como el BASIC, demuestra las posibilidades que tiene este lenguaje, tanto en versatilidad como en recursos de programación. Actualmente y sentados cómodamente delante de un teclado de un computador personal, podemos hacer que la electrónica realice en pocos minutos el trabajo que hace siglos tuvieron que hacer en 20 años, y además con seguridad en la exactitud en los resultados.



Con los computadores personales pueden seguirse, desde el propio domicilio, las evoluciones de los negocios siempre que el computador esté conectado a la red que distribuye la información desde una central de datos. (Cortesía: Thomson).

GLOSARIO DE TERMINOS ANGLOSAJONES UTILIZADOS EN INFORMATICA Y MICROCOMPUTADORES

ALU: Unidad aritmético lógica.
ACCESS CONTROL: Controles de acceso.

CONNECT TIME: Tiempo de conexión.
CPU: Unidad Central de Proceso
CYCLE TIME: Tiempo de un ciclo.
DATA BASE: Base de datos.
DBMS: Sistema de gestión de una base de datos.
DATA COMMUNICATION EQUIPMENT: Equipo de comunicación de datos.
DEBUG PROGRAMS: Programas de ayuda al programador para eliminar errores y omisiones.
DIAGNOSTIC PROGRAMS: Programas de diagnóstico.
DIRECT ADDRESSING: Direccionamiento directo.
DMA: Acceso directo a memoria.
EBCDIC: Código de intercambio decimal codificado binario extendido.
ERROR CONTROL: Control de error.



La programación de computadores es un buen ejercicio para desarrollar en los estudiantes sus cualidades.

ACCESS TIME: Tiempo de acceso.
ACCUMULATOR: Registro acumulador.
ADDRESS: Dirección de memoria.
ADDRESSING MODES: Modos de direccionamiento.
ASYNCHRONOUS OPERATION: Operación asíncrona.
ASCII: Código estándar americano para intercambio de información.



Computador Toshiba de diseño compacto, provisto de una pantalla de cristal líquido a color de alto brillo. Es de funcionamiento silencioso, sin ruidos procedentes de la unidad de disco o del ventilador de refrigeración.

BANDWIDHT: Ancho de banda de las frecuencias de transmisión de la información.
BAUD: Velocidad de transmisión de una comunicación, en bits por segundo incluyendo bits de START y STOP.
BIT: Abreviatura de dígito binario.
BREAKPOINT: Punto de ruptura de un programa. Es usado para localizar errores.
BUS: Grupo de conductores que permiten el intercambio de palabras entre la memoria la CPU, las entradas/salidas, etc.

BYTE: Secuencia de *n* bits. El más frecuente es el byte de 8 bits.

BUFFER: Sincronizador entre las velocidades de transmisión y recepción de distintos sistemas.

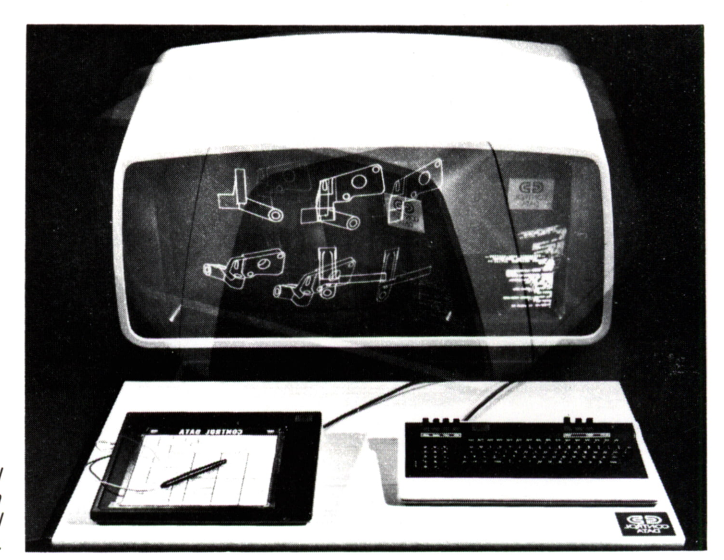
CATHODO-RAY TUBE (CRT): Tubo de rayos catódicos.

CHANNEL: Canal de información.

CLOCK: Circuito oscilador que proporciona la base de tiempo.

CODE: Código.

COMPUTER NETWORK: Conjunto de sistemas de computador interconectados formando una red.



Una pantalla móvil permite una buena observación por el programador.

FETCH: Búsqueda de una instrucción.

FIRMWARE: Es el software que se incorpora sobre las ROM de los computadores.

FORTRAN: Lenguaje de programación de alto nivel generalmente usado para usos científicos y cálculos técnicos. Proviene de la contracción de "FORmula TRANslator".

HARDWARE: Parte física que constituye un computador.

HIGH-LEVEL LANGUAGE: Lenguaje de alto nivel.

IC: Circuito integrado.

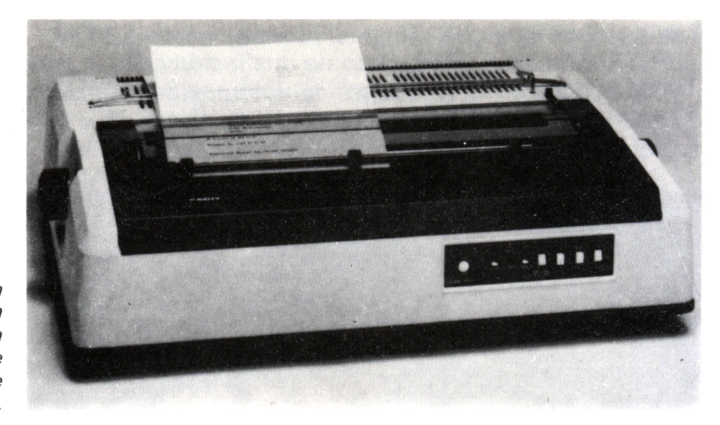
INMEDIATE ADDRESSING: Direccionamiento inmediato.

INPUT/OUTPUT (I/O): Entrada/Salida.
INSTRUCTION CYCLE: Ciclo de una instrucción.
INSTRUCTON SET: Juego de instrucciones de un procesador.



Los teclados de los computadores son, generalmente, de tipo alfanumérico, para poder trabajar con números y letras como exigen las operaciones comerciales, archivo, etc.

LOADER: Cargador.
LOW-LEVEL LANGUAGE: Lenguaje de bajo nivel.
MACHINE LANGUAGE: Lenguaje máquina.
MEMORY ADDRESS REGISTER: Registro de direcciones de memoria.
MNEMONICS: Nombres simbólicos o abreviaciones usados para definir las instrucciones, registros, memorias, etc. Técnico que permite mejorar la capacidad de la memoria humana.
OBJECT PROGRAM: Programa objeto.
OP CODE: Código de operación.
PACKET: Grupo de bits incluyendo elementos de datos y de control que debidamente transmitidos forman una información completa.



La impresora es un periférico que proporciona una gran ayuda en la programación, a la vez que permite una permanente salida de datos.

PAGE: Página. Se refiere a la organización de las memorias.

PC: Circuito Impreso.

PERIPHERAL DEVICE: Dispositivo periférico.

PLA: Red lógica programable.

PROGRAM COUNTER (PC): Contador de programa.

PROM: Memoria programable de sólo lectura.

RAM: Memoria de acceso aleatorio.

REAL TIME SYSTEM: Sistema de tiempo

RELATIVE ADDRESSING: Direccionamiento relativo.

RETOURN ROUTINE (SUBROUTINE): Subrutina.

SWAPPING: Técnica de almacenamiento de programas de baja prioridad o datos en almacenamiento secundario, de forma que el dato es únicamente extraído y accesible cuando se necesita.

TELEPROCESSING: Concepto de tranmisión y proceso de datos mediante un computador situado a distancia.

VIRTUAL ADDRESS: Dirección virtual.

WORD LENGT: Número de bits de las palabras que usan los computadores. Cuanta mayor longitud tiene la palabra, más precisión dispone la información que se transmite.

Obviamente la lista de asertos que se han introducido en este glosario no pretende ser el elemento sustitutivo de ninguna obra especializada al respecto, más bien trata de ser un útil auxiliar de consulta al que el lector pueda recurrir cuando, por ejemplo, en la lectura de un catálogo o texto técnico encuentre referencias en inglés que no sepa descifrar.

